On the Detection of Missing-Gate Faults in Reversible Circuits by a Universal Test Set

Hafizur Rahaman[1], *Dipak K. Kole*[1], *Debesh K. Das*[2], *Bhargab B. Bhattacharya*[3]

[1]IT Dept., Bengal Engg. & Sc. University, Howrah – 711 103, India; email: *rahaman_h@it.becs.ac.in*
[2]Dept. of Comp. Sc. & Engg., Jadavpur University, Kolkata – 700 032, India; email: *debeshd@hotmail.com*
[3]ACM Unit, Indian Statistical Institute, Kolkata – 700 108, India; email: *bhargab@isical.ac.in*

ABSTRACT

Logic synthesis with reversible circuits has received considerable interest in the light of advances recently made in quantum computation. Implementation of a reversible circuit is envisaged by deploying several special types of quantum gates, such as k-CNOT. Newer technologies like ion trapping or nuclear magnetic resonance are required to emulate quantum gates. Although the classical stuck-at fault model is widely used for testing conventional CMOS circuits, new fault models, namely, single missing-gate fault (SMGF), repeated-gate fault (RGF), partial missing-gate fault (PMGF), and multiple missing-gate fault (MMGF), have been found to be more suitable for modeling defects in quantum k-CNOT gates. In this paper, it is shown that in an $(n \times n)$ *reversible circuit implemented with k-CNOT gates, addition of only one extra control line along with duplication each k-CNOT gate yields an easily testable design, which admits a universal test set of size* $(n+1)$ *that detects all SMGFs, RGFs, and PMGFs in the circuit.*

Keywords: Missing-gate faults, quantum computing, reversible logic, testable design, universal test set

1. INTRODUCTION

Reversible logic can be employed to design information lossless circuits, and therefore, has the potential of reducing power consumption drastically [1-4]. It provides a basis for the newly emerging paradigm of quantum computing [5-7]. Since quantum gates or circuits satisfy "no-cloning" behavior, and are information lossless, they do not permit fanout, and ought to have an equal number of inputs and outputs. An n-input, m-output Boolean function F is said to be reversible if and only if $m = n$, and F is one-to-one.

A reversible combinational circuit must be fanout free, acyclic, and should consist of only reversible gates, which themselves implement reversible functions; such gates need to be specially designed, e.g., by using Toffoli gates. Reversible circuits have numerous applications to optical computing, digital signal processing, communication, cryptography, nanotechnology, quantum computing, DNA technology, and low-power CMOS design [5-13]. Conventional logic gates such as AND, OR, or EXOR used in digital design are not reversible. To design a reversible circuit, only reversible gates can be used, for example, the controlled-not (CNOT) gate proposed by Feynman [14], Toffoli [16], or Fredkin [15] gates. Many techniques for the synthesis of reversible logic circuits are known [17-23, 35].

Recently, several researchers [24, 27-30, 32] have studied the problem of fault modeling and testing of reversible logic circuits. The online testability in reversible circuits was studied [25-26, 29]. Universal testability of reversible logic circuits designed with k-CNOT gates under the stuck-at fault model (both single and multiple) has also been investigated [30]. A test generation scheme detecting bridging faults in reversible circuits is also reported [32, 33]. However, new fault models, namely, single missing-gate fault (SMGF), repeated-gate fault (RGF), partial missing-gate fault (PMGF), and multiple missing-gate fault (MMGF), have been found to be more suitable for modeling defects in quantum k-CNOT gates [27, 34]. They capture better representation of physical failures in reversible logic, particularly for quantum technologies. A k-CNOT based circuit can be implemented using trapped-ion technology, where the ions interact with laser pulses [5, 6, 34].

Determination of a universal test set for fault detection in reversible circuits has been studied for a few fault models [27, 30]. It has been shown that by adding one extra control line and a few 1-CNOT gates, any reversible circuit designed with k-CNOT gates can be tested for all SMGFs just by applying one test vector. All the irredundant RGFs are also detectable by the same test.

In this paper, we investigate the problem of detecting PMGFs by a universal test set. A PMGF in a k-CNOT gate may be of first or higher order depending on the number of partially misaligned or mistuned gate pulses in its quantum implementation [34]. We show that it is always possible to transform an $(n \times n)$ reversible circuit implemented with k-CNOT gates, by adding only one extra control line and by duplicating each k-CNOT gate,

so that the modified design becomes easily testable; it then admits a universal test set of size (n +1) that detects all PMGFs of any order, in the circuit. Since the test set is universal, no test generation by ATPG (Automatic Test Pattern Generation) is required. The original functionality of the circuit can be restored by setting the extra control line to logic 0. In addition, all the SMGFs, and detectable RGFs are also tested by the same test set.

2. PRELIMINARIES

Reversible logic: A reversible function has equal number of inputs and outputs, and simply induces a permutation on the set of input vectors to produce an output vector. Therefore, given an input vector, its output vector is unique, and for an output vector, its corresponding input vector can be uniquely restored. Further, in a circuit implementation with reversible gates, no fanout is allowed. In conventional non-reversible logic design, the above restrictions are not imposed. However, a non-reversible Boolean function can always be implemented by a reversible circuit after appropriately transforming it to a reversible one by adding garbage lines and reversible gates [35].

Example: The function $\{x,\ y\rightarrow\ x.y\}$ denoting AND operation is not reversible. By adding one extra input and two outputs, a modified but reversible function $\{x,\ y,\ z\ \rightarrow\ x,\ y,\ z \oplus x.y\}$ can be constructed. The AND function can be realized at the output $z \oplus x.y$, by setting the input z to constant zero; the circuit will have two "garbage" outputs. The Toffoli gate [16] realizes this function. By *garbage* is meant the number of extra outputs required to realize the given function.

Reversible gates: The basic CNOT type reversible gates used for synthesis are the following: (i) (1 × 1) NOT ($x_1 \rightarrow \bar{x}_1$); (ii) (2 × 2) controlled NOT (CNOT) gate: (x_1, x_2)→(x_1, $x_1 \oplus x_2$); and (iii) (3 × 3) Toffoli gate (x_1, x_2 , x_3)→(x_1 , x_2, $x_1x_2 \oplus x_3$).

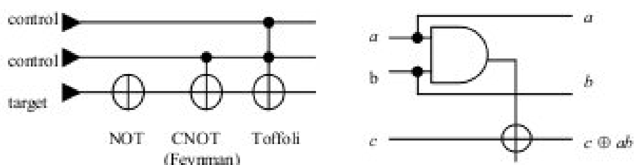

Fig. 1a: NOT, CNOT and Toffoli gates

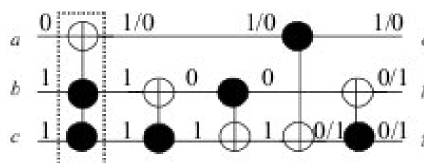

Fig. 1b: Behavior of a Toffoli gate

A generalized Toffoli gate has a set of control inputs C, a target input set T, and has the form $TOF(C;T)$, where $C = (x_{i1}, x_{i2},\ldots, x_{ik})$ $T = \{x_j\}$ and $C \cap T = \varnothing$. It maps an input vector $(x^0_1, x^0_2,\ldots, x^0_n)$ to $(x^0_1, x^0_2,\ldots, x^0_{j-1}, x^0_j \oplus (x^0_{i1}.x^0_{i2}.\ldots.x^0_{ik}), x^0_{j+1},\ldots, x^0_n)$. Thus, a NOT gate is ($TOF\ (x_j)$), a generalized Toffoli gate which has no controls. The CNOT gate is ($TOF\ (x_i;\ x_j\ )$), a generalized Toffoli gate with one control bit [14, 21]; this is also known as the Feynman gate. The simple (3 × 3) Toffoli gate is a generalized Toffoli gate with two controls [16]. These three gates are shown in Fig 1. A k-CNOT gate has k control inputs x_1, x_2, …, x_k and one target input t. It maps the input vector (x_1, x_2, …, x_k, t) to the output vector (x_1, x_2, …, x_k, $t \oplus x_1 . x_2 \ldots x_k$). In other words, a k-CNOT gate has k+1 inputs and k+1 outputs; the first k outputs follows the respective inputs, and it inverts the target at the (k+1)-th output if and only if all the k control inputs are 1. Any reversible function can be realized as a cascade of k-CNOT gates.

3. FAULT DETECTION IN A REVERSIBLE CIRCUIT

Testing of a reversible circuit, in general, turns out to be relatively simpler compared to that of non-reversible logic because of the inherent *ease of controllability* of logic states and *observability* of errors [1]. Another important property that expedites the test generation process is the fact that backtracing is straightforward and always yields a unique vector at the input.

3.1 Fault model

Several new fault models for k-CNOT based reversible circuits were introduced earlier [27, 34]. These are single missing-gate fault (SMGF), the repeated gate fault (RGF), partial missing-gate fault (PMGF), and the multiple missing-gate fault (MMGF). In this section, we briefly explain with examples, the nature of these faults.

Single missing-gate fault (SMGF): This model corresponds to the case when one k-CNOT gate completely disappears from the circuit. In the presence of this fault, the CNOT gate behaves as a simple wire connection, i.e., the pulse implementing the gate operation is short, missing, misaligned or mistuned.

Fig. 2: Single missing gate fault (SMGF)

Fig. 2 shows an SMGF marked by the dotted box in the reversible circuit ham3\design#1 benchmark, where the first 2-CNOT gate is missing.

An SMGF is detected by setting logic 1 value on all the control inputs of the gate, and any value either 0 or 1 on the target input as well as on the wires not connected to the gate. In the example of Fig. 2, if we apply {a, b, c} = {0, 1, 1} at the input of the circuit, the normal output would be {e, f, g} = {1, 0, 0}, whereas, in the presence of the SMGF fault marked by the dotted box, the output will be {e, f, g} = {0, 1, 1}. The number of possible SMGFs is equal to the number of gates in the circuit.

Repeated-gate fault (RGF): A repeated-gate fault (RGF) is an unwanted replacement of a *k*-CNOT gate by several instances of the same gate [34]. An RGF may be needed to model the occurrence of long or duplicated pulses. Fig. 3 shows an example, where first gate is repeated in the circuit ham3\design#1. The effect of this fault is thus same as that of an SMGF at the first 2-CNOT gate in the original circuit.

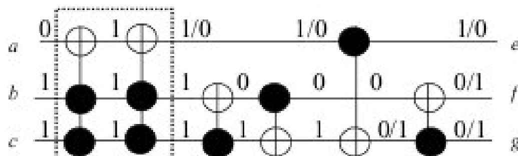

Fig. 3: RGF in ham3\design#1 reversible circuit

If we apply {a, b, c} = {0, 1, 1}, the normal output would be {e, f, g} = {1, 0, 0}, whereas, in the presence of the above RGF marked by the dotted box, the output will be {e, f, g} = {0, 1, 1}. Hence, it is detected by the vector {a, b, c} = {0, 1, 1}.

It is clear that if a RGF replaces a gate by even number of instances of the same gate, its effect is similar to the effect of the SMGF with respect to the same gate. If the RGF replaces a gate by odd number of instances of the same gate, the fault is redundant, i.e., it does not change the function of the circuit. Further, it has been shown that any SMGF test set detects all detectable RGFs [34].

Partial missing-gate fault (PMGF): This is used to model the defects resulting from the partially misaligned or mistuned gate pulses [34]. It changes a *k*-CNOT gate into a *p*-CNOT gate, with $p < k$. The corresponding fault is called as *$(k - p)^{th}$ order* PMGF. Fig. 4 shows a first-order PMGF affecting the second control input of the leftmost gate. An SMGF can be seen as a 0-order PMGF.

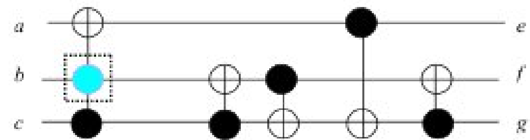

Fig. 4: PMGF in ham3\design#1

In the circuit of Fig. 4, if we apply {a, b, c} = {1, 0, 1}, the normal output would be {e, f, g} = {1, 1, 0}, whereas, in the presence of the first order PMGF fault as shown, the output will be {e, f, g} = {0, 0, 1}. Hence, the vector {a, b, c} = {1, 0, 1} detects this fault. It has been shown [34] that a PMGF (of first or higher order) is detected when a 0 is applied to at least one of the affected control inputs and a 1 to all other control inputs. Thus, a higher order PMGF is detected by a test vector for a first order PMGF, the affected control input of which is one of those affected in the higher order PMGF. Hence, it is sufficient to consider first order PMGFs only.

Multiple missing-gate fault (MMGF): This is defined as complete disappearance of two or more *consecutive* *k*-CNOT gates from the circuit.

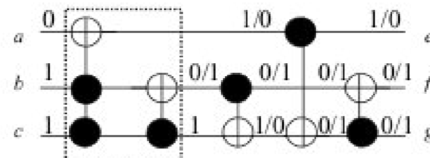

Fig. 5: MMGF in *ham3\design#1*

In the circuit of Fig. 5, it is shown that the circuit has an MMGF marked by the dotted box. This fault is detected by the vector {a, b, c} = {0, 1, 1}.

3.2 Testable design for detecting PMGFs

An exact ATPG scheme has been reported earlier [34] that generates test vectors for various types of missing-gate faults discussed above. To detect all PMGFs by a universal test, the original reversible circuit is augmented by adding one wire and duplicate *k*-CNOT gates.

A first-order PMGF affecting the jth control input can be detected by setting 0 at the jth control input and 1 at all the other control inputs. For such a vector, the fault-free and the faulty gate will produce different values on the target node. Therefore, to detect all first order PMGFs in a *k*-CNOT gate as shown in Fig. 6a, we will have to apply the following *k* test vectors $\{x_1\, x_2 \ldots x_j \,..\, x_k \,..\, t\}$: (0 1… 1…1 .. X), (1 0… 1…1 .. X), …., (1 1… 1…0 .. X) at the input level, where X, applied to the target input, may be 0 or 1.

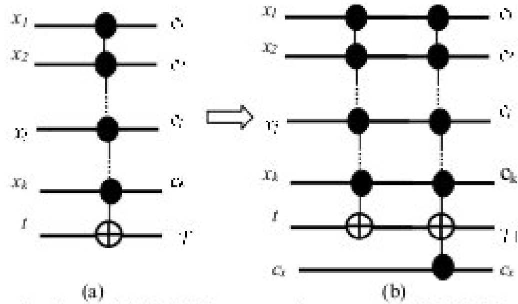

Fig.6: (a) A k-CNOT gate (b) Augmented CNOT gate.

An $(n \times n)$ reversible circuit R of depth d is built with a cascade of k-CNOT gates. While the above k test vectors applied at the inputs to R are guaranteed to detect all PMGFs at the first CNOT gate, they may not detect a PMGF at a CNOT gate lying at a subsequent level, as the vectors change when they propagate through various levels. However, if we are able to produce the same k patterns at the inputs of each CNOT gate lying at all other levels, then all PMGFs of first order can be detected in the reversible circuit. To restore the same test patterns at each level, we augment a k-CNOT gate as shown in Fig. 6b. The same k-CNOT gate is repeated consecutively, and one additional control input (c_x) is added.

Lemma 1: The target output T_1 of the augmented gate is equal to the target input t when $c_x = 1$.

Proof: The output of the target line $T = t \oplus (x_1.x_2\ldots x_j\ldots x_k)$. After augmentation, the target output T_1 when $c_x = 1$, is given by:

$$\begin{aligned} T_1 &= T \oplus (1.\ x_1.x_2\ldots x_j\ldots x_k) \\ &= T \oplus (x_1.x_2\ldots x_j\ldots x_k) \\ &= t \oplus (x_1.x_2\ldots x_j\ldots x_k) \oplus (x_1.x_2\ldots x_j\ldots x_k) \\ &= t \end{aligned}$$

Hence the proof follows.

Therefore, it is possible to restore the same test pattern (which is applied at the input level), at the output level. Repeating this augmentation procedure for every k-CNOT gate with a common additional control line (c_x), an $(n \times n)$ reversible circuit (Fig. 7) is modified as in Fig. 8.

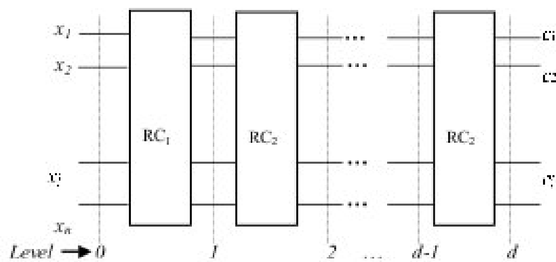

Fig. 7: An $(n \times n)$ reversible circuit of depth d

In other words, a CNOT gate is inserted between every j^{th} and $(j+1)^{th}$ level, where the inserted gate is the same as the one at the jth level with one extra control input.

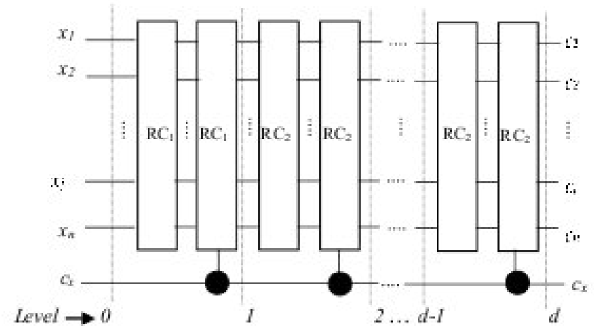

Fig. 8: Augmented reversible circuit

The augmented circuit implements the normal function when the control line c_x is set t 0.

Example: The circuit ham3\design#1 is shown in Fig. 9. The augmented circuit is shown in Fig. 10. For this circuit, the test set: $S\{a, b, c, c_x\}$ = {0111, 1011, 1101, 1110}, detects all possible first order PMGFs. Since first order PMGFs dominate all other higher order PMGFs [34], this test set detects all PMGFs in general. Further, this is universal in the sense that for all (3×3) reversible circuits, the same test set will work.

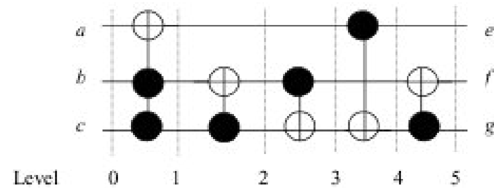

Fig. 9: ham3\design#1 benchmark reversible circuit

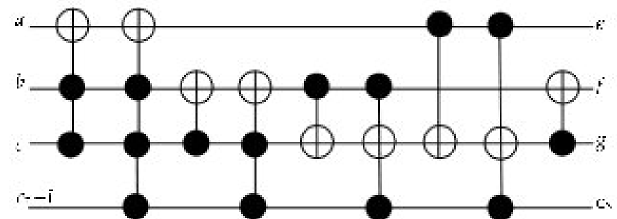

Fig. 10: Testable design for ham3\design#1

The general result stated below, now easily follows.

Theorem 1: In the testable design as shown in Fig. 8, the following universal test S_U of length $(n+1)$ is sufficient to detect all PMGFs of any order ≥ 1.

$$S_U = \begin{pmatrix} x_1 & x_2 & \ldots\ldots & x_n & c_x \\ 0 & 1 & \ldots\ldots & 1 & 1 \\ 1 & 0 & \ldots\ldots & 1 & 1 \\ \ldots & \ldots & \ldots\ldots & \ldots & \ldots \\ 1 & 1 & \ldots\ldots & 0 & 1 \\ 1 & 1 & \ldots\ldots & 1 & 0 \end{pmatrix}$$

Lemma 2: The test set S_U is also sufficient to detect all the SMGFs in the circuit.

Proof: In a k-CNOT gate, if we apply pattern $\{x_1\ x_2 \ldots, x, \ldots\ x_k\ t\} = (1\ 1\ ..1 \ldots 1\ X)$, where X denotes don't care, the target output T becomes the complement of t. Since the reversible circuit is implemented with only k-CNOT gates, only one of the n input lines of the reversible circuit is the target line t for a given CNOT gate. If we apply 0 on the target line and 1 on the remaining lines, then it is able to detect an SMGF on that gate. Clearly at the input level, such a test pattern belongs to the set S_U. For other CNOT gates at all subsequent levels, the required vector reappears in the testable design of Fig. 8, as discussed earlier. Hence, the test set S_U is sufficient to detect all SMGFs in the augmented circuit.

Lemma 3: All detectable RGFs are detected by the test set S_U.

Proof: Follows from the fact that any SMGF test set detects all detectable RGFs [34].

Thus, it follows that the above universal test S_U of length (n+1) is sufficient to detect all SMGFs, all detectable RGFs, and all PMGFs in the augmented reversible circuit. The test set depends only on n and is independent of the functionality of the reversible logic.

4. EXPERIMENTAL RESULTS

We have studied several examples of reversible benchmark circuits [36], the results of which are shown in Table 1. Column 1 shows the circuit name, and column 2 denotes gate count (N), column 3 presents the input size (n). The number of tests for detecting all PMGFs obtained by running the ATPG [34], is shown in column 4. The size of the universal test set as per the proposed method is just (n+1) and is shown in column 5. The universal test set can be directly found without the need of running an ATPG. However, the augmentation procedure doubles the gate cost.

Table 1: Comparison of the test set for the PMGF model

Circuit	N	n	# of tests as in [34]	# of tests as in the proposed method
2of5d1	18	6	8	7
4_49tc1	16	4	3	5
hwb4tc	17	4	3	5
hwb5tc	56	5	9	6
hwb6tc	126	6	15	7
hwb7tc	291	7	24	8
rd53d1	12	7	8	8
rd53rcmg	30	7	8	8

5. CONCLUSIONS

This paper presents a design-for-testability technique for testing missing-gate faults in a reversible circuit. The technique derives a universal test set of length (n+1) for detecting all partial missing-gate faults (PMGF) along with all single missing-gate faults (SMGF), and all detectable repeated-gate faults (RGF). in an ($n \times n$) reversible combinational circuit designed with k-CNOT gates. The test set also detects a large number of multiple missing-gate faults (MMGF). However, for detection of all MMGFs, additional tests and/or further augmentation may be needed. These would require further investigation.

References

1. R. Landauer, "Irreversibility and heat generation in the computing process," *IBM Research and Development*, vol. 5, pp. 183-191, 1961.
2. C. H. Bennett, "Logical reversibility of computation," *IBM J. Research and Development*, vol. 17, pp. 525-532, Nov. 1973.
3. C. H. Bennett, "Notes on the history of reversible computation," *IBM J. Research and Development*, vol. 32, pp. 16-23, January 1988.
4. R. P. Feynman, *Feynman Lectures on Computation*. (A. J. G. Hey and R. W. Allen, Ed.), Perseus Books, USA, 1996.
5. M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
6. N. Gershenfeld and I. L. Chuang, "Quantum computing with molecules," *Scientific American*, June 1998.
7. J. Preskill, *Lecture Notes in Quantum Computing*. Technical Report, (http://www.Theory.caltech.edu/~preskill/ph229).
8. W. C. Athas and L. J. Svensson, "Reversible logic issues in adiabatic CMOS," *Manuscript*.
9. P. Picton, "Optoelectronic, multivalued, conservative logic," *International Journal of Optical Computing*, vol. 2, pp. 19-29, 1991.
10. P. Picton, "A universal architecture for multiple-valued reversible logic," *MVL Journal*, vol. 5, pp. 27-37, 2000.

11. R. C. Merkle and K. E. Drexler, "Helical logic," *Nanotechnology*, vol. 7, pp. 325-339, 1996.
12. R. C. Merkle, "Reversible electronic logic using switches," *Nanotechnology*, vol. 4, pp. 21-40, 1993.
13. R. C. Merkle, "Two types of mechanical reversible logic," *Nanotechnology*, vol. 4, pp. 114-131, 1993.
14. R. Feynman, "Quantum mechanical computers," *Optics News*, vol. 11, pp. 11-20, 1985.
15. E. Fredkin and T. Toffoli, "Conservative logic," *Int. J. of Theoretical Physics*, vol. 21, pp. 219-253, 1982.
16. T. Toffoli, "Reversible computing," Tech memo - MIT/LCS/TM 151, MIT Lab for Comp. Sci., 1980.
17. V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Transactions on CAD*, vol. 22, pp. 723-729, June 2003.
18. K. Iwama, Y. Kambayashi, and S. Yamashita, "Transformation rules for designing CNOT-based quantum circuits," In *Proc. Design Automation Conference*, June 2002.
19. A. Mishchenko and M. Perkowski, "Logic synthesis of reversible wave cascades," In *Proc. International Workshop on Logic Synthesis*, pp. 197-202, June 2002.
20. M. Perkowski, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, X. Song, A. Al Rabadi, L. Joswiak, A. Coppola, and B. Massey, "Regularity and symmetry as a base for efficient realization of reversible logic circuits," In *Proc. International Workshop on Logic Synthesis*, pp. 245-252, 2001.
21. A. Mischenko and M. Perkowski, "Reversible Maitra cascades for single output functions," In *Proc. International Workshop on Logic Synthesis*, pp. 197-202, 2002.
22. D. M. Miller, "Spectral and two-place decomposition techniques in reversible logic," In *Proc. Midwest Symposium on Circuits and Systems*, Aug. 2002.
23. D. M. Miller and G. W. Dueck, "Spectral techniques for reversible logic synthesis," In *Proc. 6th International Symposium on Representations and Methodology of Future Computing Technologies*, pp. 56-62, March 2003.
24. K. N. Patel, J. P. Hayes, and I. L. Markov, "Fault testing for reversible circuits," In *Proc. VTS*, pp. 410-416, 2003.
25. J. C. Bertrand, N. Giambiasi, and J. J. Mercier, "Sur la recherche de l'inverse d'un automate," *RAIRO*, pp. 64– 87, Apr. 1974.
26. J. C. Bertrand, J. J. Mercier, and N. Giambiasi, 'Sur la recherche de l'inverse d'un circuit combinatoire," *RAIRO*, pp. 21–44, Jul. 1974.
27. J. P. Hayes, I. Polian, B. Becker, "Testing for missing-gate faults in reversible circuits," In *Proc. Asian Test Symposium*, pp. 100-105, 2004.
28. K. Ramasamy, R. Tagare, E. Perkins and M. Perkowski, "Fault localization in reversible circuits is easier than for classical circuits," In *Proc. of ATS 04*.
29. D. P. Vasudevan, P. K. Lala and J. P. Parkerson, "A novel approach for on-line testable reversible logic circuit design," In *Proc. Asian Test Symposium*, 2004.
30. A. Chakraborty, "Synthesis of reversible circuits for testing with universal test set and C-testability of reversible iterative logic arrays," In *Proc. VLSI Design*, pp. 249-254, 2005.
31. N. Scott, Reversible Circuit Viewer v1.14 Simulation.
32. H. Rahaman, D. K. Kole, D. K. Das, and B. B. Bhattacharya, "Detection of bridging faults in a reversible circuit", In *Progress in VLSI Design and Test* (Ed. C. P. Ravikumar), Elite Publishing, New Delhi, 2006, pp. 384-392.
33. H. Rahaman, D. K. Kole, D. K. Das, and B. B. Bhattacharya, "Minimal test set for bridging fault detection in reversible circuits", In *Proc. Asian Test Symp.*, Bejing, China, pp. 125-128, 2007.
34. I. Polian, J. P. Hayes, T. Fienn and B. Becker, "A family of logical fault models for reversible circuits", In *Proc. Asian Test Symp.*, Kolkata, India, 2005, pp. 422-427.
35. D. Maslov, *Reversible Logic Synthesis*, Ph.D Thesis, The University of New Brunswick, Canada, September 2003.
36. D. Maslov, G. Dueck, and N. Scott, *Reversible Logic Synthesis Benchmarks Page*. www.cs.uvic.ca/~dmaslov/, 2004.