A Simple Approach for Designing Online Testable Reversible Circuits

N. M. Nayeem and J. E. Rice
Dept. of Math & Computer Science
University of Lethbridge
Lethbridge, AB, Canada
{noor.nayeem, j.rice}@uleth.ca

Abstract

This paper presents a simple technique to convert an ESOP-based reversible circuit into an online testable circuit. The technique does not require redesigning the whole circuit for integrating the testability feature, and no new garbage outputs are produced other than the garbage outputs needed for the ESOP-circuit. With a little extra hardware cost, the resultant circuit can detect online any single-bit errors. Experimental results show that the proposed technique can achieve an improvement of up to 58% in quantum cost and 99% in garbage outputs in average, compared to the previous work.

1. Introduction

Nowadays, reversible logic is gaining popularity because of its characteristics to dissipate nearly zero energy. Traditional logic loses information during computation, causing a certain amount of heat dissipation [11]. This amount of heat is small at present, but it will become significant in the near future if Moore's law remains true. Bennett showed that a circuit consisting of only reversible gates dissipates zero energy [4]. According to Frank [8], reversible logic can recover a fraction of energy that can reach up to 100%. As there is no limit in reducing the heat dissipation in reversible logic, the amount of dissipated heat will become very close to zero. Research on reversible computing is useful for various technologies such as quantum computing [1], low power CMOS design [3], optical computing [19], nanotechnology [16], and bioinformatics.

A good amount of research work has been carried out in the area of reversible logic testing. However, there is not much work in the literature on reversible logic with online testability, except for the designs in [23, 22, 12, 13, 6, 10]. In this paper, we present a simple but efficient approach to construct online testable circuits which can detect any single-bit errors. Our technique works with the ESOP-based reversible logic synthesis described in [7], and hence no new synthesis approach is required. The general idea of our proposal is to add gates and a parity line so that any single-bit fault is propagated both forward to the end of the circuit as well as "down" to the parity line. Thus by examining a single line at the end of the circuit, any single-bit fault can be detected.

2. Background

2.1. Reversible logic

A reversible gate has the same number of inputs and outputs, mapping each input vector into a unique output vector. A reversible circuit is built using only reversible gates which are interconnected without feedback and fan out [21].

The most popular reversible gate is the Toffoli gate. An n-bit Toffoli gate, given in Figure 1(a), maps the input vector [k_1, k_2, ... , k_n] to the output vector [o_1, o_2, ..., o_n], where o_j = k_j (for j = 1, 2, ..., n-1) and o_n = $k_1k_2...k_{n-1} \oplus k_n$. The first ($n$-1) bits are known as controls. The last bit is the target which is toggled only if all of the control lines are 1. Another name for a 2-bit (that is, $n = 2$) Toffoli gate is the CNOT gate. A negative-control Toffoli gate has one or more negative controls; in that case, the target bit is toggled if all positive controls have the value 1 and the negative controls have the value 0. A 3-bit Toffoli gate with a single negative control in its first input is shown in Figure 1(b).

The extended Toffoli gate (ETG) is a multi-target Toffoli gate proposed in [5]. For our proposed design, we need a (n+1)-bit ETG with two targets (o_n and o_{n+1}) as shown in Figure 1(c). This gate has the input vector [k_1, k_2, ... , k_n, k_{n+1}] and the output vector [o_1, o_2, ..., o_n, o_{n+1}], where o_j = k_j (for j = 1, 2, ..., n-1), o_n = $k_1k_2...k_{n-1} \oplus k_n$, and o_{n+1} = $k_1k_2...k_{n-1} \oplus k_{n+1}$.

Gate count is the simplest way to evaluate different reversible circuits. If the functionally and size of the gates used for synthesis are different, then gate count cannot give

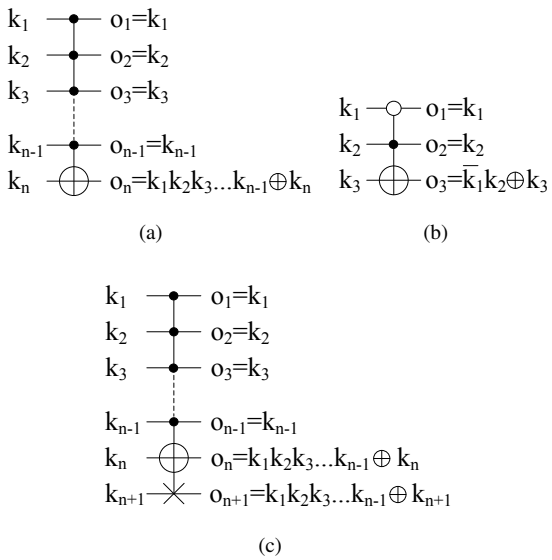

Figure 1. (a) An n-bit Toffoli gate, (b) a 3-bit negative-control Toffoli gate, and (c) a (n+1)-bit ETG.

accurate measure [17]. The quantum cost is the most effective cost metric in this regard since it refers to the number of elementary (quantum) gates required to implement the circuit. The quantum cost of a reversible circuit is the summation of the quantum costs of its gates. To calculate the quantum cost, we use the costs of gates given in [14]. Costs of a Toffoli gate and a negative-control Toffoli gate with at least one positive control are exactly the same. However, if all controls are negative, an extra cost of 2 is required. For more information about the quantum cost, please refer to [15, 2, 6]. Note that the cost of a (n+1)-bit ETG is 2 + the cost of an n-bit Toffoli gate.

In reversible circuits, some outputs are required to maintain the reversibility property but behave neither as the final results nor are they used for further calculations. These outputs are called garbage outputs. A challenging task in designing a reversible circuit is to minimize the number of garbage outputs.

Since the existing online testable designs and our proposed design use different types of gates, we evaluate the circuits in terms of quantum cost and garbage outputs.

2.2. ESOP-based synthesis

The ESOP-based reversible logic synthesis proposed in [7] works with the ESOP representation of a function and uses only Toffoli gates to design the circuit. The resultant circuit has p input lines and q output lines, where p is number of input variables and q is the number of output variables of a function, and the input lines passing through the Toffoli gates remain unchanged. The synthesis is very simple. The circuit initially has an empty cascade with $p + q$ lines. A Toffoli gate is cascaded for each ESOP term of each output. In other words, every term is mapped into a Toffoli gate. For example, given a function in the ESOP form, $f = abc \oplus cd$, a 4-bit Toffoli gate and a 3-bit Toffoli gate are required to realize the function as shown in Figure 2. The controls of the first Toffoli gate are the input lines a, b, and c; its target line is the output line f. Similarly, for the second Toffoli gate, controls are c and d; the target line is f. Note that if the ESOP term contains at least one variable in its complemented form, we then use the negative-control Toffoli gate as suggested by [20].

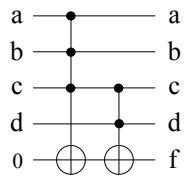

Figure 2. An ESOP-based reversible circuit.

3 Related work

Researchers have worked on various topics of reversible logic testing, such as fault modeling [9], test pattern generation [18], and many other areas. Here, we review the implementation of testable reversible circuits as reported in the literature.

In [23, 22], Vasudevan *et al.* proposed a design methodology to construct a reversible circuit with online testability. Three 4×4 reversible gates R1, R2, and R were proposed. The first gate R1 is used to realize NAND, OR, EXOR, and XNOR operations by setting different values on inputs. One of the outputs of R1 is the parity output. In R2, the first three outputs are exactly same as its first three inputs, whereas the last output is the parity outputs. In order to construct a testable block, the gates R1 and R2 are cascaded by connecting the first three outputs of R1 to the first three inputs of R2. Thus a testable block contains two parity outputs. These two parity outputs are then compared to determine whether a fault has occurred or not. The testable blocks are used to realize the reversible circuit. A checker circuit, built using R gates, is also required to test the parities of all the blocks. The R gate [22], proposed as R3 gate in [23], is a new 3×3 gate that can be used both to invert and duplicate a signal.

The authors in [12, 13] extend the above idea to easily

convert a given circuit into an online testable circuit. An improved version of checker circuit was also proposed.

In a dual rail coding approach [6], a set of 4×4 dual rail reversible gates were proposed for online fault detection. Each dual rail gate has two pairs of inputs. Two inputs of each pair are given in dual rail form, *i.e.* two inputs that are the complement of each other (either 01 or 10). If the outputs appear in dual rail form, then there is no error. However, a non-dual rail form (either 11 or 00) represents a single fault. These dual rail gates are cascaded to generate the testable circuit. A fault in the circuit propagates to the end of the circuit. Thus the fault is detected by checking the outputs of the circuit. As a result, no checker circuit is required to test the intermediate gates.

Recently, Kole *et al.* [10] proposed an online-testing technique to detect a single missing gate. Although this approach does not produce any garbage outputs, the quantum cost for adding the testability feature is quite high.

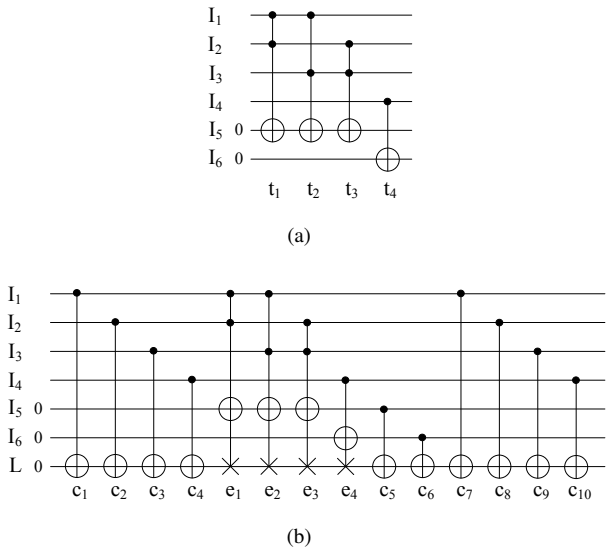

Figure 3. (a) An ESOP-based circuit, (b) Online testable reversible circuit.

4 Our Approach

Consider an ESOP-based circuit which is generated from a function with p input variables and q output variables. For designing an online testable circuit from a given ESOP-based circuit, we need some CNOT gates and a parity line L which is initialized by a 0. The procedure is as follows:

Every n-bit Toffoli gate in the given circuit is replaced by a (n+1)-bit ETG. The connections of the first n bits of the ETG remain the same as that of n-bit Toffoli gate. The last, *i.e.* ($n + 1^{st}$) bit of the ETG is connected to L. After replacing all Toffoli gates, CNOT gates are inserted from all the output lines to the L line, requiring q more gates. In order to test the input lines, we add CNOT gates from each of the input lines to L before and after the whole circuit. This step requires $2p$ CNOT gates. Now in the resulting circuit, if a single fault occurs in any of the input lines, output lines or even in L, the value of L will be changed to 1. If no fault occurs, L will remain 0. It is important to note that this technique can also be applied for the ESOP-based circuit consisting of inverted-control Toffoli gates.

This process of converting a circuit can be best described by an example. For a given 4-input (I_1, I_2, I_3, I_4), 2-output (I_5, I_6) ESOP-based circuit shown in Figure 3(a), the corresponding online testable circuit generated by the proposed technique is shown in Figure 3(b). We can see that Toffoli gates (t_1, t_2, t_3, and t_4) are replaced by ETGs (e_1, e_2, e_3, and e_4). CNOT gates c_1-c_{10} are added to test the input and output lines.

4.1 Analysis

This section proves that the proposed technique can detect any single-bit faults. Lemma 1 shows the characteristics of the ETG, which is used in Lemma 2 to prove the correctness of our technique. Two examples are also given to illustrate the propagation of a fault and the detection of such fault at the end of the circuit.

Lemma 1: Consider an (n+1)-bit ETG (see Figure 1(c)) which maps the input vector [k_1, k_2, ..., k_{n-1}, k_n, k_{n+1}] to the output vector [o_1, o_2, ..., o_{n-1}, o_n, o_{n+1}], where o_j = k_j (for j =1, 2, ..., n-1), o_n = $f \oplus k_n$, o_{n+1} = $f \oplus k_{n+1}$, and f = k_1k_2 ... k_{n-1}. If an error in any input bit between the first and ($n - 1^{st}$) bit affects the last two outputs, then o_n = $\overline{f} \oplus k_n$ and o_{n+1} = $\overline{f} \oplus k_{n+1}$. If an error occurs in the n^{th} bit, then o_n = $f \oplus k_n \oplus 1$ and o_{n+1} = $f \oplus k_{n+1}$. ◁

Proof: Consider a fault in k_j, for j = 1, 2, ..., n-1. This fault affects the calculation of function f only if k_m = 1, $\forall\ m \in \{1, 2, ..., n\text{-}1\}$ - $\{j\}$. If this condition holds, then the fault has impact on the last two outputs as both outputs compute f. Thus o_n = $\overline{f} \oplus k_n$ and o_{n+1} = $\overline{f} \oplus k_{n+1}$.

Now, assume a fault occurs in the n^{th} bit; *i.e.* in k_n. This fault does not propagate to o_{n+1} since o_{n+1} is independent of k_n. However, due to the fault, o_n = $f \oplus \overline{k}_n$ = $f \oplus k_n \oplus 1$. ◀

Lemma 2: If any single fault occurs on any line, the value of L changes to 1 and the fault is detected. ◁

Proof: Consider an online testable circuit generated by the proposed technique which has N ETGs, p input lines, q output lines, and a parity line L. Let G = $\{g_1, g_2, ..., g_N\}$ be the set of ETGs used in the circuit, and let I_1, I_2, ..., I_p be the input lines and I_{p+1}, I_{p+2}, ..., I_{p+q} be the output lines. All output lines and L are initialized by 0. Given an (n+1)-bit ETG $g_i \in G$ and the input vector [k_1, k_2, k_3, ...,

k_{n-1}, k_n, k_{n+1}], the output vector is [k_1, k_2, k_3, ..., k_{n-1}, $fg_i \oplus k_n$, $fg_i \oplus k_{n+1}$], where $fg_i = k_1k_2...k_{n-1}$, and n can be at most p+1. In order to prove this lemma, consider the following three cases:

Case 1: Assume that a single fault occurs on any input line, say I_z (for z = 1,2,...,p), which affects a set of gates, X = $\{x_1, x_2, ..., x_u\} \subseteq G$. Consider another set of gates, Y = $\{y_1, y_2, ..., y_v\}$ which is not affected by the fault. We have $G = X \cup Y$, and X or Y can be empty.

As described before, the last two outputs of a gate $g_i \in G$ computes the same function fg_i. A gate y_r in Y computes fy_r, for r = 1, 2, ..., v. However, from Lemma 1, due to the fault, each gate x in X computes $\overline{fx}_s$, for s = 1,2,...,u.

After the last gate in G, the line I_z has the faulty value $\overline{I}_z$, and L is $I_1 \oplus I_2 \oplus$ ... $\oplus I_z \oplus$ ... $\oplus I_p \oplus fy_1 \oplus fy_2 \oplus$ ... $\oplus fy_v \oplus \overline{fx}_1 \oplus \overline{fx}_2 \oplus$ ... $\oplus \overline{fx}_u$. When all output lines are EXORed to L at the end, L becomes

$I_1 \oplus I_2 \oplus ... \oplus I_z \oplus ... \oplus I_p \oplus fy_1 \oplus fy_2 \oplus ... \oplus fy_v \oplus \overline{fx}_1 \oplus \overline{fx}_2 \oplus ... \oplus \overline{fx}_u \oplus fy_1 \oplus fy_2 \oplus ... \oplus fy_v \oplus \overline{fx}_1 \oplus \overline{fx}_2 \oplus ... \oplus \overline{fx}_u$

$= I_1 \oplus I_2 \oplus ... \oplus I_z \oplus ... \oplus I_p$

Finally, when all input lines are EXORed to L, the fault on I_z is propagated to L, and L becomes

$I_1 \oplus I_2 \oplus ... \oplus I_z \oplus ... \oplus I_p \oplus I_1 \oplus I_2 \oplus ... \oplus \overline{I}_z \oplus ... \oplus I_p$

$= I_z \oplus \overline{I}_z = 1$.

Since at the end, the line L contains 1, the circuit can detect the fault.

Case 2: Now consider that a single fault occurs on an output line I_{p+z} at any point, where z = 1, 2, ..., q. From Lemma 1, gates, which have connections with line I_{p+z} after the occurrence of the fault, have the faulty values in its next to last output, but these gates produce the fault free values on L. For example, a (n+1)-bit ETG will produce the faulty value in o_n due to the fault on I_{p+z} but fault free value in o_{n+1}.

Consider a set of gates, X = $\{x_1, x_2, ..., x_u\} \subseteq G$ which have connections with line I_{p+z} after the fault occurs. Consider another set of gates, Y = $\{y_1, y_2, ..., y_v\}$ such that Y = $G - X$. Each gate x_s in X computes fx_s, for s = 1,2,...,u. Similarly, each gate y_r in Y computes fy_r, for r = 1,2,...,v.

The fault on I_{p+z} propagates to the end of the line. Using Lemma 1, the value of the line I_{p+z} at the end is $fx_1 \oplus fx_2 \oplus$ ... $\oplus fx_u \oplus 1$. As a result, when output and input lines are EXORed to L at the end, the faulty value of I_{p+z} appears at L. Thus L becomes

$I_1 \oplus I_2 \oplus$ ... $\oplus I_p \oplus fx_1 \oplus fx_2 \oplus$ ... $\oplus fx_u \oplus fy_1 \oplus fy_2 \oplus$ ... $\oplus fy_v \oplus I_1 \oplus I_2 \oplus$ ... $\oplus I_p \oplus fx_1 \oplus fx_2 \oplus$ ... $\oplus fx_u \oplus fy_1 \oplus fy_2 \oplus$ ... $\oplus fy_v \oplus 1 = 1$

Case 3: A fault can also occur on L. This causes L to have the value 1.

Hence, for any of these cases, the circuit detects the fault. ◄

The following two examples describe how this technique can detect a single fault.

Example 1: For a given 4-input (I_1, I_2, I_3, I_4), 2-output (I_5, I_6) online testable circuit shown in Figure 4, consider a single fault on input line I_2 just before the first ETG. For the faulty lines, output values are given in the form [fault-free value/ faulty value] after each gate. For other lines, only the fault free values are shown. As can be seen from the figure, the fault on I_2 causes another fault on output line I_5. However, the fault is detectable since the value of L changes to 1. This example illustrates that the circuit is able to detect a fault even though it propagates to multiple lines.

Example 2: For a given 4-input (I_1, I_2, I_3, I_4), 2-output (I_5, I_6) online testable circuit shown in Figure 5, consider a single fault on output line I_5 between the first and second ETGs. Notice that the fault propagates to the end of I_5. Due to the fault, L becomes 1 and hence the fault is detected.

5 Experimental results

A number of benchmark circuits have been collected from [24]. Table 1 compares our proposed approach to the earlier reported approaches [23, 22, 6, 12, 13] in terms of quantum cost and garbage outputs. In the table, columns QC and GO represent the quantum cost and the number of garbage outputs, respectively.

It can be seen from the table that our approach achieves a huge reduction in quantum cost and garbage outputs for every circuit. Our design produces 21.1 garbage outputs on average, whereas the best reported approach [12, 13] requires 799.9 (see Table 1). The average minimization of garbage amount is 99%, 98%, and 97% with respect to [23, 22], [6], and [12, 13], respectively.

We compute the improvement in quantum cost to be on average 50% (over [23, 22]), 58% (over [6]), and 21% (over [12, 13]).

6 Conclusion

The proposed online testable reversible design has several advantages over the existing designs. The designs in [23, 22, 6] require new synthesis approach to redesign the non-testable circuit in order to make it testable whereas our technique is rather simple since it works on top of the ESOP-based circuit, adds some CNOT gates and replaces the Toffoli gates with ETGs. Unlike [23, 22, 12, 13], our design does not need any checker circuit. One important feature of our design is that gates added for testing the circuit do not produce any new garbage. Garbage outputs of our circuit come from the ESOP-based circuit, which are negligible comparing with the previous work. Experimental results also show the efficiency of our design in reducing the quantum cost (up to 58% reduction in average). In future, online detection of multiple errors will be investigated.

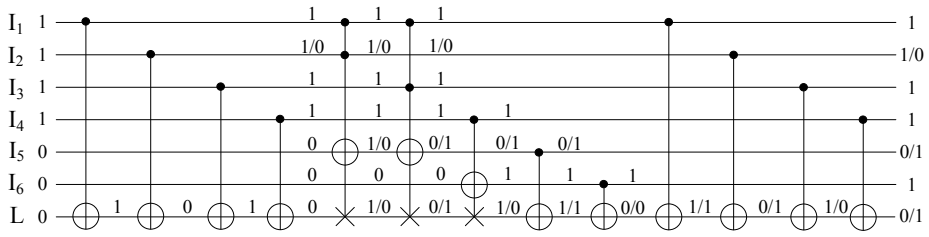

Figure 4. A single fault on input line I_2.

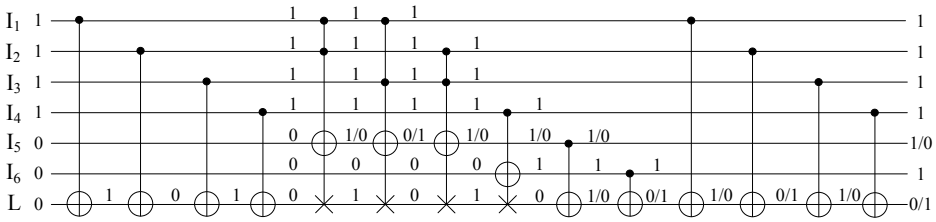

Figure 5. A single fault on output line I_5.

Table 1. Experimental results

Circuit	Approach in [23, 22]		Approach in [6]		Approach in [12, 13]		Our approach	
	QC	GO	QC	GO	QC	GO	QC	GO
9symml	25598	5952	32220	532	12148	139	11052	9
apex5	195061	45362	213843	3518	68460	1557	53917	117
apla	17815	4142	20574	334	5293	185	3954	10
bw	19019	4422	25140	386	8146	620	4847	5
cm82a	2163	502	2823	50	331	45	159	5
con1	1647	382	1929	38	287	30	180	7
cu	5560	1292	6234	110	1782	82	1326	14
dk17	10376	2412	11478	186	2385	95	1781	10
ex2	1733	402	2241	42	250	23	166	5
ex5p	84852	19732	98898	1540	37476	1518	26838	8
f51m	128927	29982	164889	2670	41842	937	33155	14
frg2	218109	50722	247332	4048	260116	5073	205890	143
max46	15321	3562	19314	326	5501	114	4627	9
misex3	141354	32872	174732	2816	148975	2987	117202	14
rd73	16482	3832	20583	342	1750	140	1052	7
sqn	11236	2612	14256	240	2917	122	2193	7
sqrt8	5689	1322	7086	122	979	76	624	8
sym9	24652	5732	31890	528	12148	139	11052	9
table3	223054	51872	265566	4294	110751	2019	87874	14
z4ml	3883	902	5070	88	1035	97	596	7
Average	57626.55	13400.5	68304.9	1110.5	36128.6	799.9	28424.25	21.1

Acknowledgement

This research was funded by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] A. N. Al-Rabadi. *Reversible Logic Synthesis: From Fundamentals to Quantum Computing*. Springer-Verlag, 2004.

[2] M. Arabzadeh, M. Saeedi, and M. Zamani. Rule-based optimization of reversible circuits. In *Proceedings of Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 849–854, 2010.

[3] W. C. Athas and L. J. Svensson. Reversible logic issues in adiabatic CMOS. In *Proceedings of Workshop on Physics and Computation(PhysComp)*, pages 111–118, Dallas, TX, 1994.

[4] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.

[5] J. Chen, X. Zhang, L. Wang, X. Wei, and W. Zhao. Extended Toffoli gate implementation with photons. In *Proceedings of 9th International Conference on Solid-State and Integrated-Circuit Technology (ICSICT)*, pages 575–578, China, 20-23 Oct 2008.

[6] N. Farazmand, M. Zamani, and M. B. Tahoori. Online fault testing of reversible logic using dual rail coding. In *Proceedings of 16th IEEE International On-Line Testing Symposium (IOLTS)*, pages 204–205, 5-7 July 2010.

[7] K. Fazel, M. Thornton, and J. E. Rice. ESOP-based Toffoli gate cascade generation. In *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 206–209, Victoria, BC, Canada, 22-24 Aug. 2007.

[8] 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, 4-6 May 2005.

[9] 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.

[10] D. K. Kole, H. Rahaman, and D. K. Das. Synthesis of online testable reversible circuit. In *Proceedings of 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pages 277–280, Vienna, 14-16 April 2010.

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

[12] S. N. Mahammad, S. Hari, S. Shroff, and V. Kamakoti. Constructing online testable circuits using reversible logic. In *Proceedings of 10th IEEE International VLSI Design and Test Symposium (VDAT)*, pages 373–383, Goa, India, August 2006.

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

[14] D. Maslov. Reversible logic synthesis benchmarks page. `http://www.cs.uvic.ca/~dmaslov/`.

[15] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne. Quantum circuit simplification and level compaction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(3):436–444, 2008.

[16] R. C. Merkle. Reversible electronic logic using switches. *Nanotechnology*, 4(1):21–40, 1993.

[17] M. Mohammadi and M. Eshghi. On figures of merit in reversible and quantum logic designs. *Quantum Information Processing*, 8:297–318, August 2009.

[18] K. N. Patel, J. P. Hayes, and I. L. Markov. Fault testing for reversible circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(8):1220–1230, 2004.

[19] P. Picton. Optoelectronic, multivalued, conservative logic. *International Journal of Optical Computing*, 2:19–29, 1991.

[20] Y. Sanaee and G. W. Dueck. ESOP-based Toffoli network generation with transformations. In *Proceedings of 40th IEEE International Symposium on Multiple-Valued Logic*, pages 276–281, 2010.

[21] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes. Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):710–722, 2003.

[22] D. P. Vasudevan, P. K. Lala, D. Jia, and J. P. Parkerson. Reversible logic design with online testability. *IEEE Transactions on Instrumentation and Measurement*, 55(2):406–414, 2006.

[23] D. P. Vasudevan, P. K. Lala, and J. P. Parkerson. Online testable reversible logic circuit design using NAND blocks. In *Proceedings of IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems*, pages 324–331, Los Alamitos, CA, USA, 10-13 October 2004.

[24] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler. RevLib: An online resource for reversible functions and reversible circuits. In *Proceedings of 38th International Symposium on Multiple Valued Logic*, pages 220–225, 2008. RevLib is available at `http://www.revlib.org`.