The Complexity of Fault Testing for Reversible Circuits

Shigeru ITO, Satoshi TAYU, and Shuichi UENO
Department of Communications and Integrated Systems, Tokyo Institute of Technology

1 Introduction

Reversible circuits, which permute the set of input vectors, have potential applications in nanocomputing [4], low power design [1], digital signal processing [7], and quantum computing [5]. It is shown in [3] that given a reversible circuit C and a set of wires F of C, it is NP-hard to generate a minimum complete test set for stuck-at faults on F. This paper shows that given a reversible circuit C, it is NP-hard to generate a minimum complete test set for stuck-at faults on the set of wires of C.

A gate is reversible if the Boolean function it computes is bijective. If a reversible gate has k input and output wires, it is called a $k \times k$ gate, or a gate on k wires. A circuit is reversible if all gates are reversible and are interconnected without fanout or feedback. If a reversible circuit has n input and output wires, it is called an $n \times n$ circuit, or a circuit on n wires.

We shall focus our attention to detecting faults in a reversible circuit C which cause wires to be stuck-at-0 or stuck-at-1. Let $L(C)$ be the set of all possible fault locations in C. $L(C)$ consists of all input and output wires of C, and input wires to gates in C. For an $n \times n$ reversible circuit C, a test is an input vector in $\{0,1\}^n$. A set of tests is said to be complete for C if it can detect all possible single and multiple stuck-at faults on $L(C)$. Patel, Hayes, and Markov showed that for any reversible circuit C, there exists a complete test set for C [6]. Let $\tau(C)$ be the minimum cardinality of a complete test set for C.

A k-CNOT gate is a reversible gate on $k+1$ wires. It passes some k inputs, referred to as control bits, to the outputs unchanged, and inverts the remaining input, referred to as target bit, if the control bits are all 1. The 0-CNOT gate is just an ordinary NOT gate. A CNOT gate is a k-CNOT gate for some k. Some CNOT gates are shown in Fig. 1, where a control bit and target bit are denoted by a black dot and ring-sum, respectively. A CNOT circuit is a reversible circuit consisting of only CNOT gates. Since the 2-CNOT gate can implement the NAND function, any Boolean function can be implemented by a CNOT circuit.

Patel, Hayes and Markov showed that

$$\tau(C) = O(\log |L(C)|)$$

for any reversible circuit C [6], and Chakraborty showed that

$$\tau(C) \leq n$$

if C is an $n \times n$ CNOT circuit with no 0-CNOT or 1-CNOT gate [2].

We show in this paper that it is NP-hard to compute $\tau(C)$ for a given CNOT circuit C. Let MTS (Minimum Test Size) be a problem of deciding if $\tau(C) \leq B$ for a given CNOT circuit C and integer B. The purpose of this paper is to prove the following:

Theorem 1 *MTS is NP-complete.* □

2 Proof Sketch of Theorem 1

To prove the theorem, we need the following characterization for a complete test set shown in [6].

Lemma I *For a reversible circuit, a test set is complete if and only if every wire can be set to both* 0 *and* 1 *by the test set.* □

MTS is in NP since a complete test set of size $O(\log |L(C)|)$ can be verified in polynomial time.

We will show a polynomial time reduction from 3SAT, a well-known NP-complete problem, to MTS. Let

$$\phi(x_1, x_2, \ldots, x_n) = \bigwedge_{i=1}^{k} P_i$$

be a Boolean function in conjunctive normal form in which each clause P_i has 3 literals for $1 \leq i \leq k$. For a Boolean variable x, literals $\overline{x}$ and x are denoted by x^0 and x^1, respectively.

We use generalized CNOT gates for simplicity. A generalized k-CNOT gate has k control bits $x_1, x_2, \ldots, x_k$ and a target bit t. The output of the target bit is defined as

$$(x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \cdots \wedge x_k^{\alpha_k}) \oplus t.$$

A control bit x_i is said to be positive if $\alpha_i = 1$, and negative if $\alpha_i = 0$. Notice that a CNOT gate is a generalized CNOT gate with no negative control bit. Notice also that a negative control bit is equivalent to a positive control bit with a 0-CNOT gate on the input and output wires.

We first construct a generalized CNOT gate G_i for each clause P_i. Let

$$P_i = x_{i1}^{\sigma_{i1}} \vee x_{i2}^{\sigma_{i2}} \vee x_{i3}^{\sigma_{i3}},$$

where $\sigma_{ij} \in \{0,1\}$ for $1 \leq j \leq 3$. We construct a generalized 3-CNOT gate G_i for P_i as follows. The gate G_i has 3 control bits x_{i1}, x_{i2}, x_{i3}, and a target bit y_i. A control bit x_{ij} is defined to be positive if $\sigma_{ij} = 0$, and negative if $\sigma_{ij} = 1$. The following lemma is immediate from the definition of G_i.

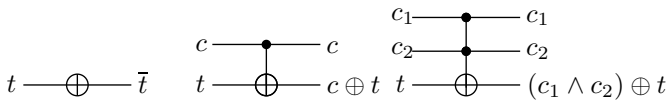
Lemma 1 *The output vector of G_i for an input vector $(x_{i1}, x_{i2}, x_{i3}, y_i)$ is $(x_{i1}, x_{i2}, x_{i3}, \overline{P_i} \oplus y_i)$.* □

Let G'_i be a copy of G_i with control bits $x'_{i1}, x'_{i2}, x'_{i3}$, and a target bit y'_i for any $i \in \{1, 2, \ldots, k\}$. For any $i, h \in \{1, 2, \ldots, k\}$, G_{ih} is a generalized 6-CNOT gate with control bits $x_{i1}, x_{i2}, x_{i3}, x'_{h1}, x'_{h2}, x'_{h3}$, and a target bit t_{ih}. A control bit $x_{ij}[x'_{hj}]$ is positive in G_{ih} if and only if $x_{ij}[x'_{hj}]$ is positive in $G_i[G'_h]$. We construct a CNOT circuit $C_1(\phi)$ on $2n+k^2$ wires which is a cascade consisting of gates G_{ih} $(1 \leq i, h \leq k)$. As an example, $C_1(\psi)$ for a Boolean function

$$\psi(x_1, x_2, x_3) = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$$

is shown in Fig. 2, where a negative control bit is denoted by an empty circle. We can prove the following by using Lemma 1.

Lemma 2 *The output vector of $C_1(\phi)$ for an input vector $(x_1, x_2, \ldots, x_n, x'_1, x'_2, \ldots, x'_n, y_1, y_2, \ldots, y_k^2)$ is also $(x_1, x_2, \ldots, x_n, x'_1, x'_2, \ldots, x'_n, y_1, y_2, \ldots, y_{k^2})$ if and only if $\phi(x_1, x_2, \ldots, x_n) = 1$ or $\phi(x'_1, x'_2, \ldots, x'_n) = 1$.* □



(a)0-CNOT gate. (b)1-CNOT gate. (c)2-CNOT gate.

Figure 1: CNOT gates.

We finally construct a CNOT circuit $C_2(\phi)$, which is obtained from $C_1(\phi)$ as shown in Fig. 3. We can prove the following by using Lemmas I and 2.

Lemma 3 *ϕ is satisfiable if and only if $\tau(C_2(\phi)) = 2$.*

Since $C_2(\phi)$ can be constructed in polynomial time, we complete the proof of the theorem.

References

[1] C. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, vol. 17, pp.525-532, 1973.

[2] A. Chakraborty, "Synthesis of Reversible Circuits for Testing with Universal Test Set and C-Testability of Reversible Iterative Logic Arrays," *Proc. of the 18th International Conference on VLSI Design*, 2005.

[3] S Ito, Y Ito, S Tayu, and S Ueno, "On the Complexity of Fault Testing for Reversible Circuits," *Technical Report of the IEICE*, Vol.105, No.387, pp.13-16, 2005.

[4] R. C. Merkle, "Two types of mechanical reversible logic," *Nanotechnology*, vol. 4, pp. 114-131, 1993.

[5] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[6] K. N. Patel, J. P. Hayes, and I. L. Markov, "Fault Testing for Reversivle Circuits," *IEEE Trans. Computer-Aided Design*, vol. 23, pp.1220-1230, Aug. 2004.

[7] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. Computer-Aided Design*, vol.22, pp. 710-722, June 2003.

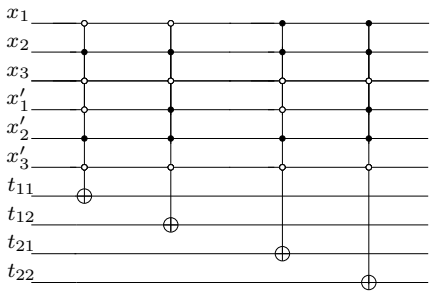


Figure 2: CNOT circuit $C_1(\psi)$.

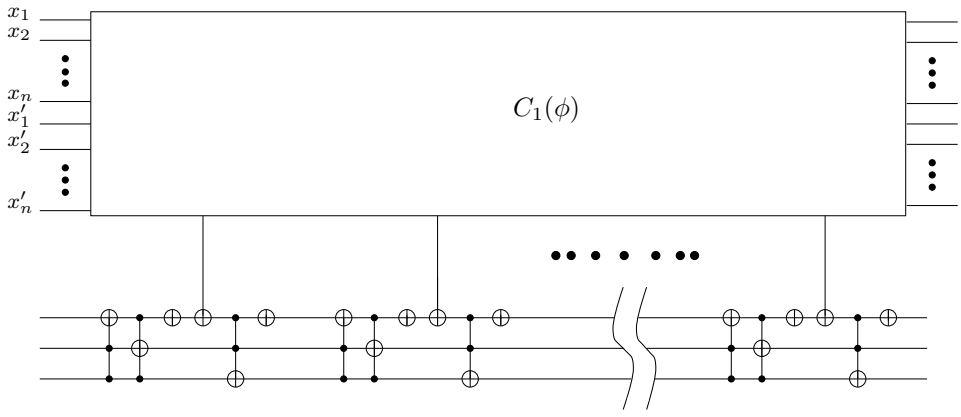


Figure 3: CNOT circuit $C_2(\phi)$