Fault Testing for Reversible Circuits[*]

Ketan N. Patel, John P. Hayes and Igor L. Markov
University of Michigan, Ann Arbor 48109-2122
{knpatel,jhayes,imarkov}@eecs.umich.edu

Abstract

Irreversible computation necessarily results in energy dissipation due to information loss. While small in comparison to the power consumption of today's VLSI circuits, if current trends continue this will be a critical issue in the near future. Reversible circuits offer an alternative that, in principle, allows computation with arbitrarily small energy dissipation. Furthermore, reversible circuits are essential components of quantum logic. We consider the problem of testing these circuits, and in particular, generating efficient test sets. The reversibility property significantly simplifies the problem, which is generally hard for the irreversible case. We discuss conditions for a test set to be complete, give a number of practical constructions, and consider test sets for worst-case circuits. In addition, we formulate the problem of finding minimal test sets into an integer linear program (ILP) with binary variables. While this ILP method is infeasible for large circuits, we show that combining it with a circuit decomposition approach yields a practical alternative.

1. Introduction

The original motivation for the study of reversible circuits is the possibility of nearly energy-free computation. Landauer [14] showed that traditional irreversible circuits necessarily dissipate energy due to the erasure of information. It was later shown that, in principle, it was possible to perform reversible computation with arbitrarily small energy dissipation [2, 8]. Though the fraction of the power consumption in current VLSI circuits attributed to information loss is negligible, this is expected to change as increasing packing densities force the power consumption per gate operation to decrease, making reversible computation an attractive alternative.

A major new motivation for the study of reversible circuits is provided by the emerging field of quantum computation [15]. In a quantum circuit the operations are performed on quantum states or qubits rather than bits. Since quantum evolution is inherently reversible, the resulting quantum computation is as well. Classical reversible circuits form an important class of these quantum circuits.

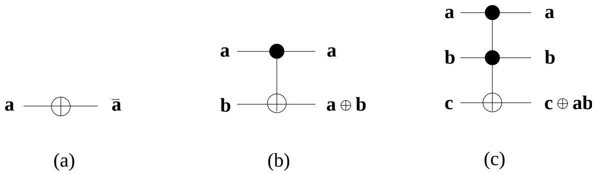

Figure 1. Examples of reversible logic gates: (a) NOT (b) C-NOT (c) Toffoli.

Very little previous research has been done on testing for reversible circuits. A notable exception is work done at Montpellier, where reversibility was studied in the context of on-line testing [3, 4]. In their work reversibility was used to synthesize on-line test structures for irreversible circuits. In contrast, our focus is on testing of inherently reversible circuits. In particular, we consider the problem of generating efficient test sets for these circuits. Though this problem is hard for conventional irreversible circuits, it can be significantly simplified in our case. Agrawal [1] has shown that fault detection probability is greatest when the information output of a circuit is maximized. This suggests that it may be easier to detect faults in reversible circuits, which are information lossless, than in irreversible ones. While this previous work focused on probabilistic testing, here we are concerned with complete deterministic testing. One of our results shows that surprisingly few test vectors are necessary to fully test a reversible circuit under the multiple stuck-at fault model, with the number growing at most logarithmically both in the number of inputs and the number of gates. This provides additional motivation for studying reversible circuits, namely they may be much easier to test than their irreversible counterparts.

2. Notation

A logic gate is *reversible* if the mapping of inputs to outputs is bijective, that is, every distinct input yields a distinct output, and the number of input and output wires are equal. If it has k inputs (and outputs), we call it a reversible $k \times k$ gate. Three commonly used gates, composing the CNT-gate library, are shown in Figure 1. The NOT gate inverts the input, the C-NOT gate passes the first input through and inverts the second if the first is 1, and the Toffoli gate passes the first two inputs through and inverts the third if the first two are both 1.

A *well-formed reversible circuit* is constructed by starting with

[*] This work was supported by the DARPA QuIST program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies of endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.

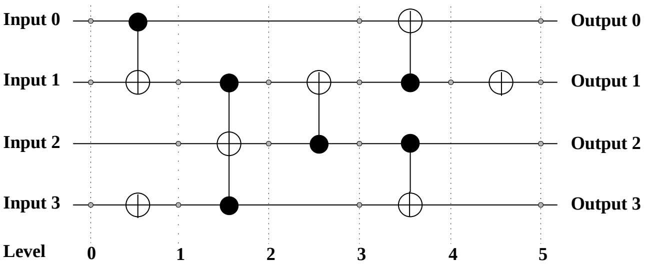

Figure 2. Reversible circuit example. The dotted lines represent the levels in the circuit, and the small open dots represent possible stuck-at fault sites.

n wires, forming the basic circuit, and iteratively concatenating reversible gates to some subset of the output wires of the previous circuit. The outputs of each reversible gate replace the wires at its input. This iterative construction naturally gives us the notion of *levels* in the circuit; the inputs to the circuit are at level 0, and the outputs of any gate are at one plus the highest level of any of its inputs. For convenience in cases where a wire at the input of a gate is at level i and the outputs are at level $j > i+1$, we say the input is at all levels between i and $j-1$ inclusively. This gives us n wires at each level. Figure 2 shows an example of a reversible circuit with the levels denoted by dotted lines. The *depth* d of the circuit is the maximum level, which can be no larger than the number of gates in the circuit. We will often find it convenient to use an n-bit vector to refer to the value of the wires at a given level in the circuit. A binary vector has *weight* k if it contains exactly k ones, and we denote the all-0's and all-1's vectors by $\underline{0}$ and $\underline{1}$ respectively.

The iterative construction also gives us the notion of a *sub-circuit*, the part of the original circuit between levels i and j, or more specifically, the circuit formed by the gates with outputs at level greater than i and less than $j+1$. We denote the function computed by the sub-circuit as $f_{i,j}$ and its inverse as $f_{i,j}^{-1}$. If we omit the first subscript i it should be assumed to be 0. The function of the entire circuit is then f_d.

We say a reversible circuit is *L-constructible*, if it can be formed using the L-gate library. Some important gate libraries used here are the CNT-gate library mentioned above, the C-NOT gate library consisting of only C-NOT gates, and the U-gate library which consists of all possible reversible $n \times n$ gates. These three gate libraries compute the set of even permutations, the set of linear reversible functions, and the set of all permutations respectively [16]. The reversible circuit shown in Figure 2 uses the CNT-gate library.

3. Complete Test Sets

The goal here is, given a reversible circuit C and a fault set F, to generate a set of test vectors that can be used to detect all faults in F. We call such a test set *complete*. A complete test set with the fewest possible vectors is *minimal*.

Two properties of reversibility simplify the test set generation problem. The first is *controllability*: there is a test vector that will generate any given desired state on the wires at any given level. The second is *observability*: any single fault that changes an intermediate state in the circuit will also change the output. Neither property holds in general for irreversible circuits.

For most of this paper we adopt the stuck-at fault model used in testing conventional circuits, which includes all faults that fix the values of wires in the circuit to either 0 or 1. For reversible circuits we show that any test set that detects all single faults, also detects any number of simultaneous faults. In Section 6 we extend our results to an alternate, and more general, model where the fault set consists of single gate failures.

3.1 General Properties

The following proposition provides a simple necessary and sufficient condition for a test set to be complete for the stuck-at fault model.

PROPOSITION 1. *Under the single stuck-at fault model a test set is complete if and only if each of the wires at every level can be set to both 0 and 1 by the test set.*

Proof Assume without loss of generality that a test set does not set a wire at level i to 0. A stuck-at 1 fault at this point in the circuit is then undetectable, since the outputs from the test set are unaffected.

On the other hand, if all wires at every level can be set to both 0 and 1 by the test set, then a stuck-at fault must affect at least one test vector, changing the value of the wire at that level from a 0 to a 1 or vice versa. By the observability property this change will affect the output. □

The following proposition shows that, in fact, the single stuck-at fault and multiple stuck-at fault models are equivalent for reversible circuits. The intuition behind this property is that in the case of multiple faults the final fault(s), i.e., those closest to the outputs, can be detected by working backwards from the outputs.

PROPOSITION 2. *Any test set, that is complete for the single stuck-at fault model is also complete for the multiple stuck-at fault model.*

Proof Suppose we have a counter-example. Then there must be a complete test set for some reversible circuit under the single fault model, which is not complete for multiple faults. So at least one multiple fault M is undetectable by our test set. Since M is undetectable, the outputs of the circuit to our test set must be the same as those of the fault-free circuit. Now M is composed of faults at various levels. Let t be the deepest level containing a sub-fault of M. Since no sub-faults occur at any level greater than t, the reversible sub-circuit between level t and the outputs is identical to the corresponding sub-circuit in the fault-free circuit. Therefore, since the outputs to our test set and the reversible sub-circuit between level t and the outputs are the same as for the fault-free circuit, the values of the wires at level t must also be the same as for the fault-free circuit. Since our test set is complete under the single fault model, by Proposition 1 each wire at level t must take both the value 0 and 1. However this is a contradiction, since there is at least one sub-fault at level t that fixes the value of a wire. □

This equivalence between the single and multiple stuck-at fault models allows us restrict our attention to the simpler case of single faults. If we have an n-wire circuit with l gates of sizes $k_1, \ldots, k_l$, then a total of $2(n + \sum_{i=1}^{l} k_i)$ single stuck-at faults can occur: stuck-at 0 and stuck-at 1 faults for each gate input and circuit output. Reversibility then implies the following lemma.

LEMMA 1. *Each test vector covers exactly half of the possible faults, and each fault is covered by exactly half of the possible test vectors.*

Proof Each test vector sets the bit at each fault site to either 0 or 1, detecting either a stuck-at 1 or stuck-at 0 fault respectively. Therefore, it detects precisely half of the possible single stuck-at faults. For a given stuck-at fault there are 2^{n-1} possible values of the bits at that level that can detect the fault, namely those that set the faulty bit to the opposite of the stuck-at value. Since the circuit is reversible, each of these can be traced back to a distinct input vector. Therefore, half of the 2^n input vectors detect the fault. □

We can obtain some properties of a minimal test set of a circuit by decomposing the circuit into sub-circuits. For example, the size of a minimal test set for a reversible circuit is greater than or equal to that of any of its sub-circuits. On the other hand, the size of a minimal test set for a circuit formed by concatenating reversible circuits $C_1, \ldots, C_k$ is no greater than the sum of the sizes of minimal test sets for the individual C_i's. Finally if two reversible circuits C_1 and C_2, with minimal test sets with sizes $|T_1|$ and $|T_2|$ respectively, act on a disjoint set of input/output bits, then the size of the minimal test set of the circuit formed by concatenating C_1 and C_2 is equal to $\max\{|T_1|, |T_2|\}$. These properties can be used to bound the size of the minimal test set, and in some cases, to simplify the problem of finding a minimal test set.

3.2 Test Set Construction

The following proposition gives a number of complete test set constructions implicitly providing upper bounds on the size of the minimal test set.

PROPOSITION 3. *A complete test set for an n-wire circuit with depth d and a total of l gates with sizes $k_1, \ldots, k_l$ is given by:*

a. any $2^{n-1}+1$ distinct test vectors

b. the following $d+2$ test vectors

$$\left\{\underline{0},\ \underline{1},\ f_1^{-1}(\overline{f_1(\underline{0})}), \ldots,\ f_d^{-1}(\overline{f_d(\underline{0})})\right\}$$

c. some set of $\left\lfloor \log_2\left(n+\sum_{i=1}^{l} k_i\right)\right\rfloor + 2$ test vectors.

Proof
(a) The value of a wire at a given level is set to 0 (or 1) by exactly 2^{n-1} input vectors. Therefore, if the test set contains $2^{n-1}+1$ vectors, then at least one will set it to 1 (or 0). Since this is true for all fault sites, by Proposition 1 the test set is complete.
(b) The vector $f_i^{-1}(\overline{f_i(\underline{0})})$ sets the wires at level i to the bitwise inverse of the values set by the $\underline{0}$ vector. Therefore each wire at every level can be set to both 0 and 1 by the test set. By Proposition 1 the test set is complete.
(c) To prove this part we first prove that given a reversible circuit and an incomplete set of test vectors, there is a test vector that can be added that covers at least half of the remaining faults.

Let m be the number of test vectors given, F_C be the faults covered by this set, and C the size of F_C. If none of the remaining $2^n - m$ input vectors cover at least half of the remaining faults, then they must each cover more than half of the faults in F_C. By Lemma 1 every test vector covers exactly $n+\sum_{i=1}^{l} k_i$ faults and every fault is covered by exactly 2^{n-1} test vectors. Therefore, the number of times faults in F_C are covered (by all input vectors cumulatively) is $2^{n-1}\cdot C$. Therefore, we have the following inequalities:

$$\begin{aligned}(2^n-m)\left(\frac{C}{2}\right) &< 2^{n-1}\cdot C - m\left(n+\sum_{i=1}^{l} k_i\right)\\ 2\left(n+\sum_{i=1}^{l} k_i\right) &< C\end{aligned}$$

The second inequality is false since the number of faults covered cannot be larger than the total number of faults that can occur. Therefore we have a contradiction, and there must be a test vector that can be added to cover at least half of the remaining faults.

By recursively applying this observation we can reduce the number of uncovered faults to 0 in no more than

$$\left\lfloor \log_2\left(n+\sum_{i=1}^{l} k_i\right)\right\rfloor + 2$$

steps (test vectors). □

Proposition 3 limits the size of the minimal test set based on the size of the reversible circuit both in terms of its depth and the number of input/output bits; for the circuit in Figure 2, parts a-c of the proposition give upper bounds of 9, 7, and 6 test vectors, respectively. The final part of the proposition implies that a reversible circuit can be tested by a very small set of tests. As an example, a reversible circuit on 64 wires with a million 3 × 3 gates can be tested using no more than 23 input vectors. However, while the first two parts of the proposition give practical constructions, the last one does not; consequently, it may not be easy to find such a test set.

4. (L,n)-Complete Test Sets

We define a test set as (L,n)-*complete* for gate library L acting on n wires, if it is complete for all circuits formed by the library. The following proposition shows that there is a circuit that requires such a test set.

PROPOSITION 4. *Any reversible gate library L acting on n wires has an (L,n)-complete set of test vectors that is minimal for some circuit in the set.*

Proof Let $C_1, \ldots, C_N$ be a set of circuits that computes the set of all functions computable using L, and $C = C_1 C_1^{-1} \cdots C_N C_N^{-1}$. Then any test set that is complete for C must be complete for any circuit formed by L. Therefore, a minimal test set for C is (L,n)-complete. □

The following proposition characterizes (L,n)-complete test sets for three classes of reversible circuits: C-constructible, U-constructible, and CNT-constructible.

PROPOSITION 5.

a. A (C,n)-complete test set must have at least $n+1$ vectors. One such set is given by the all-0's vector and the n weight-1 vectors.

b. A (U,n)-complete test set must have at least $2^{n-1}+1$ vectors, and any $2^{n-1}+1$ test vectors will give such a set.

c. A (CNT,n)-complete test set must have at least $2^{n-1}+1$ vectors, and any $2^{n-1}+1$ test vectors will give such a set.

Proof

(a) Any input to the circuit can be written as a linear combination of the n weight-1 vectors. Furthermore, since the gate library is linear (under the operation of bitwise XOR), the corresponding values of the wires at the ith level can be written as the same linear combination of the values for these weight-1 vectors. If any input vector sets the value of a wire at the ith level to 1, then so must at least one weight-1 vector. Since there are inputs that do, the weight-1 vectors are sufficient for setting all wires to 1. Furthermore, since the circuit is linear the all-0's vector sets all wires at all levels to 0. Therefore, this is a (C,n)-complete test set.

On the other hand, if the test set consists of only n input vectors, we have two possibilities: either the set spans the n-dimensional space or it does not. If the latter is true, a linear reversible circuit can be constructed that maps the test set into the $(n-1)$-dimensional subspace $0X\cdots X$, implying that the test set is not complete. If the test set spans the entire n-dimensional space, we can construct a linear reversible circuit that maps them to the following linearly independent vectors:

$$\begin{array}{rcccccc} v_1 \rightarrow & 1 & 0 & 0 & 0 & \cdots & 0 \\ v_2 \rightarrow & 1 & 1 & 0 & 0 & \cdots & 0 \\ v_3 \rightarrow & 1 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & & \vdots \\ v_n \rightarrow & 1 & 1 & 1 & 1 & \cdots & 1 \end{array}$$

Since the first wire cannot be set to 0 the test set is not complete for this circuit.

(b) Suppose we have a (U,n)-complete test set with 2^{n-1} test vectors. Because the gate library computes all permutations, we can generate a circuit mapping all 2^{n-1} test vectors to output vectors of the form 0XX$\cdots$X. This test set does not set the first output bit to 1, and thus is not complete for this U-gate circuit. This implies it is not (U,n)-complete. By Proposition 3a any $2^{n-1}+1$ test vectors will give (U,n)-completeness.

(c) Any permutation can be composed from a series of transpositions. The CNT gate library can construct circuits computing any even permutation of the input values [16], that is, a permutation that can be composed from an even number of transpositions. Following the proof for part b, a permutation can map any 2^{n-1} test vectors to output vectors of the form $0XX\cdots X$. If this permutation is even we have shown that this is an incomplete test set, otherwise we can add a transposition that exchanges the outputs $00\cdots 0$ and $00\cdots 1$. This new permutation is even and still maps the test vectors to the set of outputs $0XX\cdots X$, and therefore, the test set is not complete for this CNT-circuit. By Proposition 3a any $2^{n-1}+1$ test vectors will give (CNT,n)-completeness. □

5. ILP formulation

While Proposition 3c guarantees that an efficient test set exists for any reversible circuit, it gives no practical construction. In this section, we formulate the problem of constructing a minimal test set as an integer linear program (ILP) with binary variables. We then use this to find a practical heuristic for generating efficient test sets.

Alternately we could formulate the problem as an instance of *satisfiability* (SAT) [9]. This is distinct from the SAT formulation of the ATPG (automatic test pattern generation) problem [17], where the object is to find a test pattern that detects a given fault or prove the fault undetectable. In our case the ATPG problem is relatively easy because of reversibility; the difficult part is finding a minimal test set.

		Circuit Length (gates)							
Test Size	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
2	1	6	24	67	134	155	105	21	-
3	-	6	78	558	2641	8727	16854	10185	577
4	-	-	-	-	5	39	90	47	-

Table 1. Minimal test set size distribution for optimal 3-wire CNT-circuits with respect to gate count.

5.1 Basic Formulation

We can formulate the minimal test set problem as an ILP with binary decision variables t_i associated with each input vector; t_i takes a value of one if the corresponding input vector is in the test set, and zero otherwise. We ensure the completeness of the test set by a set of $2n(d+1)$ linear inequality constraints. A minimal test set is then determined by minimizing the sum of the t_i's.

$$\textbf{Minimize } t_0 + t_1 + \cdots + t_{2^n-1}$$

$$\textbf{subject to the constraints}$$

$$\sum_{i=0}^{2^n-1} f_j\left(T_i\right) \cdot t_i \geq \underline{1}$$

$$\sum_{i=0}^{2^n-1} \overline{f_j\left(T_i\right)} \cdot t_i \geq \underline{1}, \quad \text{for all } 0 \leq j \leq d$$

$$\text{where } t_i \in \{0,1\},\ 0 \leq i \leq 2^n - 1, \text{ and}$$

$$T_i \text{ is the } n\text{-bit binary expansion of integer } i$$

Each feasible solution gives a complete test set composed of those vectors i for which $t_i = 1$. For relatively small circuits this ILP can be solved efficiently using an off-the-shelf optimization tool such as CPLEX [12].

Using this formulation and CPLEX 7.0, we obtained minimal test sets for all optimal 3-wire CNT-circuits. CPLEX was able to solve the ILP for each circuit in a fraction of a second on a Sun SPARC. Table 1 gives a distribution of minimum test set size with respect to the number of gates in the circuit. We should note that the optimal CNT implementation of a given function is not unique, and therefore the distribution in Table 1 may be dependent on the particular optimal set chosen.

As expected, the number of test vectors needed generally increases with the length of the circuit. However, there are long circuits that have more efficient test sets than much shorter circuits. The largest minimal test set had 4 vectors, however, we can construct suboptimal circuits that require 5 test vectors.

5.2 Circuit Decomposition Approach

Solving the ILP exactly is feasible for small circuits; however, since the number of variables increases exponentially with the number of input/output bits, it is impractical for large circuits. An alternate approach is to decompose the original circuit into smaller

```
1)  Partition circuit into disjoint sub-circuits
    $C_0,\ldots,C_l$ each acting on $\leq m$ wires
2)  Initialize test_set $=\{\}$ and $i=0$
3)  Generate ILP for $C_i$ as in Section 5.1
4)  Add constraints for each vector in test_set
5)  Solve ILP
6)  Incorporate new test vectors into test_set,
    setting any unused wires of new vectors to
    don't cares
7)  Apply $C_i$ to test_set, setting don't cares at
    inputs of $C_i$ to 0
8)  If $i<l$, $i=i+1$ and go to Step 3
9)  Set remaining don't cares in test_set to 0
10) Apply $C^{-1}$ to test_set to get complete test set
```

Figure 3. Algorithm for complete test set generation based on circuit decomposition.

sub-circuits acting on fewer input/output bits, and use the ILP formulation iteratively for these sub-circuits combining the test vectors dynamically; a similar approach has been used for irreversible circuits [10]. While the resulting test set is not guaranteed to be minimal, it is generally small enough to enable efficient testing. Furthermore, it may be possible to use lower bounds to ensure the test set is not much larger than a minimal one. For example, the size of the minimal test set of a sub-circuit can be used to bound that of the larger circuit.

The algorithm shown in Figure 3 uses this decomposition approach. First the circuit is decomposed into a series of circuits acting on a smaller number of wires. One way to do this is to start at the input of the circuit and add gates to the first sub-circuit C_0 until no more can be added without having C_0 act on more than m wires. Then we continue with C_1, and so on until the entire circuit has been decomposed. The remaining steps in the algorithm are best illustrated by an example.

Consider the decomposition of the reversible circuit in Figure 4. Though the entire circuit acts on six wires, each sub-circuit acts on no more than four. Using the ILP formulation on C_0 gives test vectors:

$$\begin{array}{rcccccccccccccc} & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline v_0 = & X & 0 & 1 & X & 1 & 1 & \overset{C_0}{\Longrightarrow} & X & 1 & 1 & X & 0 & 0 \\ v_1 = & X & 1 & 0 & X & 0 & 0 & & X & 1 & 0 & X & 1 & 0 \\ v_2 = & X & 1 & 1 & X & 1 & 0 & & X & 0 & 0 & X & 1 & 1 \end{array}$$

where the X's represent don't cares and the left and right halves represent the test vectors at the input and output of C_0, respectively. Sub-circuit C_1 acts on wires x_0, x_1, x_4 and x_5. We generate the ILP for C_1, and add the following constraints:

$$\begin{array}{cccccccccc} x_0 & x_1 & x_4 & x_5 & & \multicolumn{5}{c}{\text{Constraints}} \\ \hline X & 1 & 0 & 0 & & t_4 & + & t_{12} & \geq & 1 \\ X & 1 & 1 & 0 & \in T \Rightarrow & t_6 & + & t_{14} & \geq & 1 \\ X & 0 & 1 & 1 & & t_3 & + & t_{11} & \geq & 1 \end{array}$$

Solving this ILP gives the solution $t_6 = t_{11} = t_{12} = 1$. Incorporating these values into the previous test vectors we have:

$$\begin{array}{ccccccccccccc} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline 1 & 1 & 1 & X & 0 & 0 & \overset{C_1}{\Longrightarrow} & 0 & 1 & 1 & X & 0 & 0 \\ 0 & 1 & 0 & X & 1 & 0 & & 1 & 1 & 0 & X & 1 & 0 \\ 1 & 0 & 0 & X & 1 & 1 & & 1 & 0 & 0 & X & 1 & 1 \end{array}$$

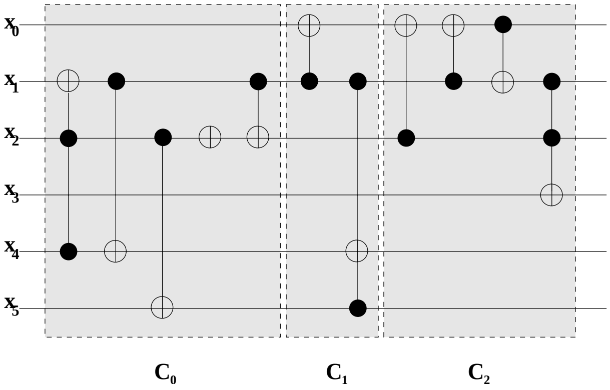

Figure 4. Circuit decomposition example.

Sub-circuit C_2 acts on wires x_0, x_1, x_2, and x_3. We generate the ILP for this sub-circuit, and incorporate the current test set using the following constraints:

$$\begin{array}{cccccccccc} x_0 & x_1 & x_4 & x_5 & & \multicolumn{5}{c}{\text{Constraints}} \\ \hline 0 & 1 & 1 & X & & t_6 & + & t_7 & \geq & 1 \\ 1 & 1 & 0 & X & \in T \Rightarrow & t_{12} & + & t_{13} & \geq & 1 \\ 1 & 0 & 0 & X & & t_8 & + & t_9 & \geq & 1 \end{array}$$

Solving this ILP gives solutions t_5, t_7, t_8, and t_{12}. The last three can be incorporated into the previous test set, however the first test vector must be added:

$$\begin{array}{ccccccccccccc} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline 0 & 1 & 1 & 1 & 0 & 0 & \overset{C_2}{\Longrightarrow} & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & X & X & & 1 & 0 & 0 & 1 & X & X \end{array}$$

Filling the don't cares with 0's and applying C^{-1} to the test set yields a complete test set for C. While the resulting test set is not guaranteed to be minimal in general, in this case it is, as can be shown by applying the ILP method on the entire circuit.

5.3 Test Set Compaction

The circuit decomposition method in the previous section will generally lead to redundant test sets. One way to reduce this redundancy is to compact the test set, that is, find the smallest complete subset. This approach has been considered previously for use with ATPG (automatic test pattern generation) algorithms for conventional circuits [11, 6, 7].

The ILP formulation in Section 5.1 can be used to perform this test set compaction. We simply eliminate all test vectors that are not in the original complete test set, along with the corresponding columns in the constraint matrix. Generally, this ILP can be solved more efficiently than the minimal test set ILP, since it has significantly fewer variables.

5.4 Simulation Results

In this section we discuss the results of simulations we have conducted to evaluate the performance of our algorithm. We generated random CNT-circuits of various lengths over 8, 16, 24 and 32 wires. The circuits were generated by selecting at random from

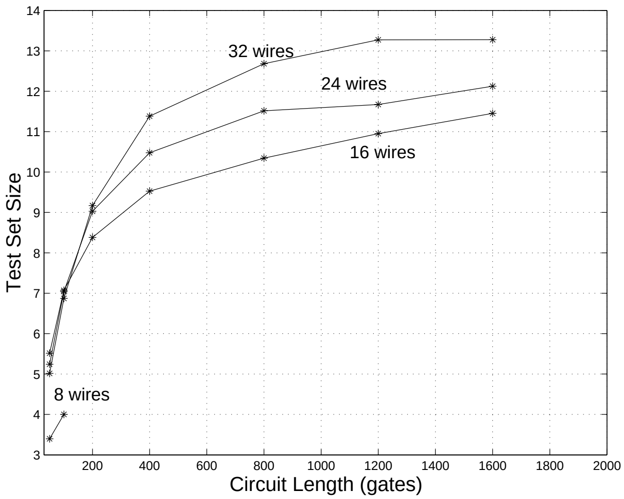

Figure 5. Average test set size vs. circuit length for circuit decomposition algorithm limiting sub-circuit size to 8 wires.

the set of all allowable NOT, C-NOT, and Toffoli gates. Each circuit was decomposed into sub-circuits acting on at most 8 wires, and our algorithm was used to find a complete test set. Figure 5 shows the average number of test vectors needed with respect to the circuit length. At least 150 circuits were generated for each data point.

The average execution time for the algorithm seemed to increase linearly with circuit length and did not vary very much with the number of input/output wires, all with the exception of the 8-wire case for which execution time appeared to increase exponentially with circuit length. This is most likely due to the fact that for this latter case the number of constraints increases linearly with the number of gates, yielding increasingly difficult ILPs. On the other hand, for the circuits on more than 8 wires, an increase in the length of circuit does not generally lead to significantly harder individual ILPs, rather only a (linearly) larger number of them to solve.

Test compaction, as expected, was most effective for longer circuits, eliminating an average of approximately one redundant test vector for circuits lengths of 800 or more.

6. Cell Fault Model

While the use of the stuck-at fault model has been very effective in conventional circuit testing, other fault models may be more appropriate for reversible circuits, especially in the quantum domain. For example, the cell fault model [13], where the function of the faulty $k \times k$ gate changes arbitrarily from the desired function, may be more realistic. In this section we extend some of our results to this model.

The following proposition provides a simple necessary and sufficient condition for a test set to be complete for the cell fault model.

PROPOSITION 6. *Under the cell fault model a test set is complete if and only if the inputs of every $k \times k$ gate in the circuit can be set to all 2^k possible values by the test set.*

Proof If a test set does not set the input wires of a gate to a particular value say $\underline{a}$, then it would not be able to detect a failure in this gate that only affects the output of $\underline{a}$.

On the other hand, if the input wires of every gate in the circuit can be set to all possible values by the test set, then any single-gate failure will affect at least one test vector, changing the value at the output of the gate. By the observability property of reversible circuits, this will be reflected in a change at the output. □

Let $g_1, \dots, g_l$ be the gates in a reversible circuit, and $k_1, \dots, k_l$ the respective gate sizes. If we consider every possible value at the input of each gate as representing a distinct fault, the total number of faults that need to be covered is $\sum_{i=1}^{l} 2^{k_i}$. Under this definition, we have the following lemma.

LEMMA 2. *Each input vector covers exactly l faults, and a fault associated with a $k \times k$ gate is covered by exactly 2^{n-k} input vectors.*

Proof Each input vector sets the bits at the inputs of each gate to some value. Therefore, since there are l gates, the vector can detect l faults. For a given fault associated with a $k \times k$ gate there are 2^{n-k} possible values for the n bits at that level that can detect it. Since the circuit is reversible, each of these can be traced back to a distinct input vector. □

The following proposition, which is analogous to Proposition 3, gives upper bounds on the size of the minimal test set under the cell fault model.

PROPOSITION 7. *A complete test set under the cell fault model for an n-wire reversible circuit with a total of l gates with sizes $k_1 \geq k_2 \geq \dots \geq k_l$ is given by*

a. any $2^n - 2^{n-k_1} + 1$ distinct test vectors

b. a set of $\left(\sum_{i=1}^{l} 2^{k_i}\right) - l + 1$ test vectors

c. some set of at most $\sum_{i=1}^{l} \left\lceil \frac{2^{k_i}}{i} \right\rceil$ test vectors

Proof
(a) For any $k \times k$ gate in the circuit there are 2^{n-k} distinct inputs that yield a particular value at its input. Therefore, if the test set has $2^n - 2^{n-k_1} + 1$ vectors (implying that fewer than 2^{n-k} are not included) then it must include at least one such input. Since this is true for all gates in the circuit, by Proposition 6, the test set is complete.
(b) Any input vector will cover l faults leaving $\sum_{i=1}^{l} 2^{k_i} - l$. By the controllability property we can cover these with one test vector each. Therefore, all of the faults can be covered with no more than $\sum_{i=1}^{l} 2^{k_i} - l + 1$ test vectors.
(c) We first prove that given an incomplete set of m test vectors covering faults in the set F_C, there must be a test vector that covers at least

$$l - \left\lfloor \sum_{f \in F_C} 2^{-k(f)} \right\rfloor \qquad (1)$$

of the remaining faults, where $k(f)$ is the size of the gate associated with fault f.

Suppose this is false. By Lemma 2 every test vector covers exactly l faults and a fault f is covered by exactly $2^{n-k(f)}$ input values. Therefore the number of times faults in F_C can be covered is $\sum_{f\in F_C} 2^{n-k(f)}$ and the current test set accounts for $m\cdot l$ of these. Furthermore, each of the remaining input vectors must cover more than $\sum_{f\in F_C} 2^{-k(f)}$ of the already covered faults, otherwise our assertion would be true. Combining these we have the following inequalities.

$$\begin{aligned}(2^n - m)\left(\sum_{f\in F_C} 2^{-k(f)}\right) &< \sum_{f\in F_C} 2^{n-k(f)} - m\cdot l \\ l &< \sum_{f\in F_C} 2^{-k(f)}\end{aligned}$$

The second inequality is false since the right side can be no larger than l. Therefore we have a contradiction, and our proposition must be true.

We can iteratively apply this property to obtain an upper bound on the number of test vectors needed for completeness. A weak form of the above result enables a simple closed form upper bound. To cover the first 2^{k_l} faults we need at most $\lceil 2^{k_l}/l\rceil$ test vectors, since each test vector we add covers l faults. To cover the next $2^{k_{l-1}}$ faults we need at most $\lceil 2^{k_{l-1}}/(l-1)\rceil$ test vectors, and so on. Thus, we can cover all single cell faults using no more than $\sum_{i=1}^{l}\lceil 2^{k_i}/i\rceil$ test vectors. □

For a reversible circuit on 64 wires with a million 3×3 gates parts a-c of Proposition 7 give upper bounds of approximately 10^{19}, $7\cdot 10^6$ and 10^6 test vectors, respectively. However, since part c uses the property illustrated in Equation 1 very conservatively in order to obtain a closed form, a much tighter bound can be obtained by applying this property directly. In fact, by iteratively applying this property, one can show that no more than 108 test vectors are needed for complete testing.

To obtain an ILP formulation for the cell fault model only the constraints given in Section 5 need to be modified. For each $k\times k$ gate at each level we generate 2^k constraints, one for each of the possible inputs to the gate. The circuit decomposition method from Section 5.2 as well as the test set compaction method in Section 5.3 can be applied as in the stuck-at fault case.

7. Conclusions and Future Work

We use the property of reversibility to simplify the testing problem for reversible circuits, and give conditions for an input set to fully test a reversible circuit under both the stuck-at and cell fault models. We develop some theoretical results on test set constructions and (L,n)-complete test sets for several reversible gate libraries. The problem of finding minimum size test sets is formulated as an integer programming problem. Because this formulation is intractable for large circuits, we give a circuit decomposition method, incorporating the ILP formulation. This approach yields complete, though not necessarily minimal, test sets. Simulation results show that the resulting test sets are generally small.

An ongoing area of work is determining good lower bounds on the size of the minimal test set. This would be useful in evaluating the efficiency of complete test sets generated by heuristic algorithms such as ours. Towards this latter goal, we also plan to search for non-trivial test circuits with known minimal test set sizes. In addition to fault detection, we also plan to study fault diagnosis, that is, using test sets to localize faults. As with fault detection, fault diagnosis may be much easier for reversible circuits than for irreversible ones. Finally, though we have focused on testing of classical reversible circuits here, we hope to extend our work to non-classical quantum circuits as well. In particular, the work using the cell fault model in Section 6 may be useful for the quantum case.

REFERENCES

[1] V. D. Agrawal. An information theoretic approach to digital fault testing. *IEEE Trans. on Comp.*, pp. 582–587, Aug. 1981.

[2] C. H. Bennett. Logical reversibility of computation. *IBM J. of Research and Development*, pp. 525–532, Nov. 1973.

[3] J. C. Bertrand, N. Giambiasi, and J. J. Mercier. Sur la recherche de l'inverse d'un automate. *RAIRO*, pp. 64–87, Apr. 1974.

[4] 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.

[5] M. L. Bushnell and V. D. Agrawal. *Essentials of Electronic Testing for Digital Memory & Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, Boston, 2000.

[6] J.-S. Chang and C.-S. Lin. Test set compaction for combinational circuits. *IEEE Trans. on CAD*, pp. 1370–1378, Nov. 1995.

[7] P. F. Flores, H. C. Neto, and J. P. Marques-Silva. On applying set covering models to test set compaction. *Proc. GLS-VLSI*, pp. 8–11, Mar. 1999.

[8] E. Fredkin and T. Toffoli. Conservative logic. *Intl. J. of Theoretical Physics*, vol. 21, pp. 219–253, 1982.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[10] P. Goel and B. C. Rosales. Test generation & dynamic compaction of test. *Dig. Papers Intl. Test Conf.*, pp. 189–192, Oct. 1979.

[11] D. S. Hochbaum. An optimal test compression procedure for combinational circuits. *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp. 1294–1299, Oct. 1996.

[12] ILOG CPLEX. http://www.ilog.com/products/cplex.

[13] W. H. Kautz. Testing for faults in cellular logic arrays. *Annual Symp. on Switching and Automata Theory*, pp. 161–174, 1967.

[14] R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. of Research and Development*, pp. 183–191, Jul. 1961.

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

[16] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes. Reversible logic circuit synthesis. *Proc. IEEE/ACM Intl. Conf. on CAD*, pp. 353–360, Nov. 2002.

[17] P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Combinational test generation using satisfiability. *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp. 1167–1176, Sept. 1996.