

A THEORY OF ERROR-RATE TESTING

Shideh Shahidi and Sandeep K. Gupta

Electrical Engineering Department, University of Southern California, Los Angeles, CA 90089
{sshahidi,sandeep}@poisson.usc.edu

Abstract— We have entered an era where chip yields are decreasing with scaling. A new concept called intelligible testing has been previously proposed with the goal of reversing this trend for classes of systems which do not require completely error-free operation. Such error tolerant applications include audio, speech, graphics, video, and digital communications. Analyses of such applications have identified error rate as one of the key metrics for error severity, where error rate is defined as the percentage of vectors for which the value at outputs deviates from the corresponding error-free value. In error-rate testing, every fault with an error rate less than a threshold specified by the application is called an acceptable fault; all other faults are called unacceptable. The objective of error-rate testing is to detect every unacceptable fault while detecting none of the acceptable faults.

In this paper we develop a theory of error-rate testing. First we study fanout-free circuits with primitive gates and identify new relationships between error rates and fault equivalence and dominance, develop a new test generation procedure, and prove that in such circuits it is possible to detect every unacceptable fault without detecting any acceptable fault. We then analyze more general circuits, including those containing complex gates and fanouts, and show that the above result may not hold for such circuits. We then use a modified version of a classical test generator and a classical fault simulator to obtain empirical data that show that even in arbitrary circuits, it is possible to detect every unacceptable fault while detecting only a fraction of acceptable faults.

Index Terms—ATPG, error rate, error tolerance, intelligible testing, test generation.

I. INTRODUCTION

VLSI scaling has entered an era where chip yields are falling [1]. At the present, this is only limiting the decrease in chip cost from one fabrication technology generation to the next. However, in a few technology generations, this phenomenon will start increasing chip costs. A new concept called *intelligible testing* was proposed in [2] with the goal of reversing this trend for classes of systems which do not require completely error-free outputs. Such *error tolerant* applications include, but are not limited to, audio, speech, graphics, video, games, and digital communications.

A number of studies have analyzed such applications. The motion detection block of an MPEG encoder [8] and the linear transform block of a JPEG encoder [9] have been studied and found to perform acceptably in presence of any one of a large

set of single stuck-at faults. This happens for two main reasons. First, some faults in these blocks do not cause errors at the block's outputs. Second, and more importantly, many other faults do cause errors at the block's outputs, but either (i) cause errors with only small significance, or (ii) cause errors with low rates. In either case, the fault causes only a slight degradation in the quality of output video and/or the level of compression.

In [10], it was shown that in transmission of coded data over a noisy channel, faults in the memory of the decoder cause errors at the output of the block with only slightly higher rates than the block error rate of the fault-free decoder, resulting in minor and acceptable performance loss. It was also shown in [11] that some faults in the memory of an answering machine lead to acceptable recording quality.

The above analyses of applications have identified two key metrics for error severity, namely *error significance* and *error rate*. For a faulty circuit version, error significance for a set of circuit outputs is defined as the maximum amount by which the response at the set of outputs can deviate from the corresponding error-free value. Error rate is defined as the percentage of vectors applied during normal circuit operation for which the value at a set of outputs deviates from the corresponding error-free value. While it is possible to define composite metrics that consider error rate in conjunction with error significance, in the following we treat the error rate metric in the form defined above.

In this paper we develop a theory for test generation for combinational circuits where we wish to only target faults which have error rates higher than or equal to a threshold. For simplicity of presentation we limit the discussion to the case where all outputs of a circuit are viewed as a single set, i.e., as a single bus. (This can be generalized in a straightforward manner to cases where circuit outputs are viewed as multiple busses and where each bus has its own threshold. It is also easy to consider special output busses, e.g., a bus containing all control outputs with a threshold value of zero.) We assume that the application has been analyzed to determine a threshold value for error rate. A fault is said to be an *unacceptable fault* if it can cause an error rate that exceeds a given threshold; otherwise the fault is said to be *acceptable*.

While a classical automatic test pattern generator (ATPG) must ideally distinguish every faulty circuit under test (CUT) from fault-free CUTs, the *primary objective* of the proposed ATPG is to distinguish any CUT which has an unacceptable fault from any CUT that is either fault-free or has an acceptable fault. In applications where a fault-free CUT commands a higher market price than a faulty CUT which has

an acceptable fault, a secondary objective is to distinguish a CUT which has an acceptable fault from a fault-free CUT. We concentrate on the new challenges posed by the primary problem, since once these challenges have been addressed the secondary problem lends itself to engineering solutions.

Other approaches have been proposed to characterize error rate in combinational CUTs [5] [6]. In [5], a classical self-test procedure has been modified to estimate the error rate for the CUT by analyzing compressed CUT responses. In [6], error rate is computed for faults and unacceptable faults are identified. A classical automatic test pattern generator (ATPG) is used to generate test vectors for unacceptable faults. This approach ignores the fact that a test vector generated for an unacceptable fault may also detect acceptable faults. Since each vector in the test set can detect a number of acceptable faults, the total number of acceptable faults covered by these test vectors may be high. In other words, in reality a large portion of chips with only acceptable faults may fail the test and hence the yield benefits of error-rate testing might be seriously compromised.

In this paper we develop a theory for error-rate testing that not only takes into account the detection of unacceptable faults but also focuses on acceptable faults. In the next section we present the key challenge posed by our primary problem. In Section III we discuss the relationships between error rate and fault equivalence and dominance. In Section IV we construct a new test generation approach that covers all unacceptable faults, i.e., maximizes test quality, without detecting any acceptable fault, i.e., maximizes yield gain of error-rate testing, for fanout-free circuits with primitive gates. Section V discusses additional challenges posed by some types of non-primitive gates and fanouts. In Section VI we experiment with benchmark circuits using a modified version of classical ATPG to empirically identify the relationship between test quality and yield gain for arbitrary circuits. Section VII discusses the conclusions and future research directions.

II. TEST OBJECTIVE AND THE CENTRAL QUESTION

A. Test Application Scenario

In classical testing scenarios which use ATPG generated vectors (also called *deterministic* vectors) for testing, automatic test equipment (ATE) is used (i) to apply each generated test to the inputs of a CUT, (ii) to capture the response at its outputs, and (iii) to compare the captured response with that expected of the fault-free version. In such testing, one vector is applied and the CUT is discarded if the response captured at CUT outputs differs from the corresponding fault-free response; otherwise, the testing continues with the next vector in the test sequence. Such a testing approach is called stop on first error (SOFE) and is attractive due to its low cost.

In the error-rate testing scenario, occurrence of erroneous response for one vector is not a sufficient condition for discarding a CUT. For example, consider a case where we test a circuit using N random vectors, generated such that the probabilities of occurrence of vectors are identical to those

during normal operation of the circuit. In such testing, CUTs with erroneous responses for up to (approximately) $T_{er}N$ vectors would be classified as acceptable. The use of a random test sequence is not practical for ATE based testing since sufficiently high fault coverage of unacceptable faults is achieved only when $N = \mathbf{O}(1/T_{er})$ vectors are applied. The number of vectors is unacceptably large in cases where T_{er} value is small. Hence, we do not consider random testing. (Random testing may be viable for built-in self-test and is studied in that context in [5].)

In this paper we study an approach for error-rate testing that uses ATPG generated, i.e., *deterministic*, test vectors to reduce N , the number of vectors applied, and hence the test cost. Since the vectors applied to CUT are not random, we cannot classify CUTs which have erroneous responses for $T_{er}N$ or fewer vectors as acceptable. For this reason as well as to reduce the test application cost, we adopt the classical SOFE test application scenario, i.e., discard a CUT if it produces erroneous response for even one vector.

B. The Central Question

The benefits of error-rate testing are maximized if (i) we maximize *test quality*, i.e., detect as many unacceptable faults as possible, **and** (ii) maximize *yield gain*, i.e., reject as few chips with acceptable faults as possible. While the requirement (i) above is encountered in classical testing, our problem is different because we must concurrently satisfy requirement (ii).

The above observation leads to the central question that is addressed in this paper: ***Is it possible to generate a set of deterministic tests that detects every detectable unacceptable fault while detecting none of the acceptable faults?***

In this paper, we develop the theory that answers the above question. Recall that every unacceptable fault f_i has error rate $R(f_i) \geq T_{er}$ and every acceptable fault f_j has error rate $R(f_j) < T_{er}$. Also, recall that the error rate for each fault depends on the probabilities with which the 2^m possible vectors are applied to circuit C during its normal operation. Most of our theoretical results are valid independent of the probabilities with which each vector occurs during normal operation. In cases where this is not true, we will explicitly state that.

We also pursue empirically a slightly less constrained version of our central question: What is the minimum number of acceptable faults that must be detected by a deterministic test set that detects every detectable unacceptable fault in the circuit? Note that we have formulated this question in a manner that does not compromise test quality.

III. BACKGROUND: RELATIONSHIPS BETWEEN FAULTS

The notions of *fault equivalence* [13] [15] and *fault dominance* [14] [15] are used extensively in fault simulation and test generation. We use these notions to identify relationships between error rates for different faults in a circuit.

Consider an arbitrary combinational circuit C . Let C^f denote a version of the circuit with fault f . Equivalence and dominance relationships between two faults f_i and f_j can be summarized in the following properties. Here, $C^f(V)$ denotes

the input-output logic behavior of C^f for vector V , and $SATV(f)$ denotes the set of all test vectors for fault f , i.e., a set containing every vector that detects f . (It is important to keep in mind that throughout this paper we consider faulty circuit versions with **either** f_i **or** f_j , i.e., we do **not** consider versions that have both f_i and f_j .)

Property 1: If two faults f_i and f_j in an arbitrary combinational circuit C are equivalent, then (i) $C^{f_i}(V) = C^{f_j}(V), \forall V$; (ii) $SATV(f_i) = SATV(f_j)$; and (iii) $R(f_i) = R(f_j)$.

Property 2: If fault f_i in an arbitrary combinational circuit C dominates f_j , then (i) $C^{f_i}(V) = C^{f_j}(V), \forall V \in SATV(f_j)$; (ii) $SATV(f_i) \supseteq SATV(f_j)$; and (iii) $R(f_i) \geq R(f_j)$.

In Properties 1 and 2, we explicitly include the relationships between $SATV(f_i)$ and $SATV(f_j)$ because if these sets satisfy the conditions of the type shown, then the corresponding relationships between error rates are satisfied for **any** distribution of the probabilities of occurrence of 2^m vectors during normal circuit operation. We include the relationships between $C^{f_i}(V)$ and $C^{f_j}(V)$ because these show that even in cases where the outputs of C are partitioned into multiple buses in arbitrary ways, the corresponding relationships between error rates are satisfied on a bus-by-bus basis.

Equivalence and dominance relationships exist between single stuck-at faults associated with every *primitive gate* (i.e., INV, NAND, AND, NOR, and OR). Every primitive gate has a *controlling value* (e.g., see [15]), cv , which is a logic value whose application to one gate input determines the logic value at the gate's output, independent of the values applied at other inputs of the gate. The *controlled response*, cr , of a gate is the logic value implied at its output by the application of its controlling value at one or more of its inputs. The complement of a gate's controlling value is called its *non-controlling value*, ncv . The concept of controlling value can be extended for many commonly used families of *complex gates*, notably AND-OR-INVs (AOI) and OAI. However, some complex gates, e.g., XOR and XNOR, do not have controlling values. For such gates, the controlling value, non-controlling value, as well as controlled response are undefined.

We can now enumerate the equivalence and dominance relationships between the single stuck-at faults associated with inputs and the output of a k -input primitive gate. Let the gate inputs be x_1, x_2, \dots, x_k and the gate output be z .

- Faults $x_1SAcv, x_2SAcv, \dots, x_kSAcv$, and $zSAcr$ are equivalent.
- Fault $zSA\overline{cr}$ dominates each of the single stuck-at faults $x_1SAncv, x_2SAncv, \dots, x_kSAncv$.

IV. ERROR-RATE TESTING OF FANOUT-FREE CIRCUITS WITH PRIMITIVE GATES

We start by re-examining and extending the above properties for *fanout-free circuits* with only primitive gates. We use the extended properties to construct a new test generator for unacceptable faults in such circuits. We then show that the vectors generated by the new test generator can

detect every unacceptable fault without detecting any acceptable fault. We consider commonly used complex gates (AOI, OAI, XOR, XNOR, and multiplexers) as well as circuits with *fanouts*, *non-reconvergent* as well as *reconvergent*, in the next section (for definition of terms see [15]).

A. Stuck-at Faults at a Primitive Gate

Consider a primitive gate in a fanout-free circuit, namely a NAND gate with inputs x and y and output z . The fact that the NAND gate is in a fanout-free circuit does not help identify any stronger relationship between the set of all test vectors and error rates for the three equivalent faults associated with the gate, namely x SA0, y SA0, and z SA1. That is, Property 1 still holds. In contrast, for the dominance relationship we can identify stronger conditions.

We exploit two special properties of fanout-free circuits (e.g., see [15]). First, in a fanout-free circuit, there is a unique path from the output of any gate to the primary output. Hence, the fault effect due to any single stuck-at fault associated with a *gate under consideration* (GUC) must be propagated along one particular path. We call this path the *unique propagation path* (UPP) and call the gates and lines along this path *on-path gates* and *on-path lines*, respectively. The effect of a fault at the GUC can only propagate to one input of each on-path gate. We call such an input of each on-path gate as the gate's *on-path input* and all its other inputs as *side-inputs*.

Fig 1 shows the above NAND gate (marked as GUC) within an arbitrary fanout-free circuit. Thick lines are used to depict the unique propagation path via which the fault effect for any fault associated with the GUC, i.e., our NAND gate, must be propagated. On-path lines are called o_1, o_2, \dots, o_k and side inputs of on-path gates are called s_1, s_2, \dots, s_l .

To detect any single stuck-at fault associated with the GUC, we must apply appropriate values (i) at the inputs of the GUC to excite the fault (and, in the case of a fault at an input of the GUC, to also propagate its effect to the output of the GUC), and (ii) at the side inputs, s_1, s_2, \dots, s_l , of the on-path gates to propagate its effect along the UPP to the primary output. In the example in Fig 1, we need to apply appropriate values at lines $x, y, s_1, s_2, \dots, s_l$.

The second special property of a fanout-free circuit is that the sub-circuits driving the lines where we need to apply appropriate values are fanout-free and pair-wise disjoint. In the example in Fig 1, sub-circuits driving lines $x, y, s_1, s_2, \dots, s_l$ are pair-wise disjoint and each sub-circuit is fanout-free.

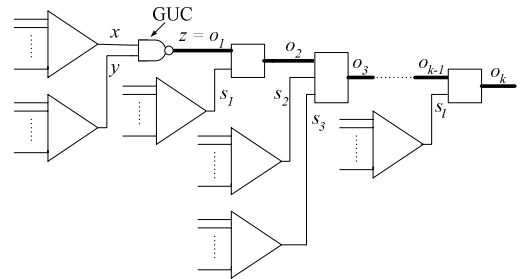


Fig 1. Testing a single stuck-at fault associated with a gate in a fanout-free circuit.

The consequence of the above properties is that it is possible to apply any combination of values at the inputs of the GUC that will excite an associated single fault (and, if the fault is at an input of the GUC, propagate its effect to the output of the GUC) and, for each such combination of values, it is possible to propagate the effect of the fault to the primary output of the circuit.

For our example two-input NAND gate, z SA0 dominates x SA1 and z SA0 dominates y SA1. Note that the fault z SA0 can be excited by applying 00, 01, or 10 at gate inputs, x and y . In contrast, we can excite x SA1 and propagate its effect to z by applying 01 at x and y . Similarly, we can excite y SA1 and propagate its effect to z by applying 10 at x and y . When viewed with the abovementioned special properties for a fanout-free circuit, we can modify the general relationship between the sets of all test vectors for these faults by replacing the ' \supseteq ' relation by the ' \supset ' relation, to obtain $SATV(zSA0) \supset SATV(xSA1)$ and $SATV(zSA0) \supset SATV(ySA1)$. Furthermore, by applying 00 at the GUC inputs, x and y , we can detect the fault z SA0 without detecting either of the faults it dominates, namely x SA1 and y SA1. This can be generalized for arbitrary primitive gates in fanout-free circuits to obtain a strengthened version of Property 2 in the following form.

Property 3: Consider a primitive gate. Let $f_o = zSA\bar{c}$ be the single stuck-at fault at the output z of a primitive gate which dominates each of the single stuck-at faults $f_1 = x_1SAncv, f_2 = x_2SAncv, \dots, f_k = x_kSAncv$ at its inputs x_1, x_2, \dots, x_k . If the primitive gate is used in a fanout-free combinational circuit C , then

- (i) $C^{f_o}(V) = C^{f_i}(V), \forall V \in SATV(f_i), \forall i \in \{1, 2, \dots, k\}$;
- (ii) $SATV(f_o) \supset \{SATV(f_1) \cup SATV(f_2) \cup \dots \cup SATV(f_k)\}$; and
- (iii) $R(f_o) > R(f_1) + R(f_2) + \dots + R(f_k)$.

An important consequence of Property 3 is that in a fanout-free circuit one or more vectors exist that can detect the fault $f_o = zSA\bar{c}$ without detecting any of the faults it dominates, namely $f_1 = x_1SAncv, \dots, f_k = x_kSAncv$. Any vector in the set $SATV(f_o) - \{SATV(f_1) \cup \dots \cup SATV(f_k)\}$ makes this possible.

For equivalent faults associated with a primitive gate with inputs x_1, x_2, \dots, x_k and output z embedded in an arbitrary fanout-free circuit, we restate Property 1 in the form of (1) and (2).

$$SATV(zSA\bar{c}) = SATV(x_1SAncv) = \dots = SATV(x_kSAncv), \quad (1)$$

and

$$R(zSA\bar{c}) = R(x_1SAncv) = \dots = R(x_kSAncv). \quad (2)$$

For the faults with dominance relationship, we restate Property 3 in the form of (3) and (4).

$$SATV(zSA\bar{c}) \supset \{SATV(x_1SAncv) \cup \dots \cup SATV(x_kSAncv)\}, \quad (3)$$

and

$$R(zSA\bar{c}) > R(x_1SAncv) + \dots + R(x_kSAncv). \quad (4)$$

Fig 2 shows an example fanout-free circuit and illustrates the relationships between the sets of all test vectors for every single stuck-at fault in the circuit. Every box with one or more fault names as a label is the set of all possible vectors for the corresponding faults. Note that for every single stuck-at fault in this circuit there exists a vector that detects the fault without detecting any fault that has a lower error rate than the target fault. For example, we can use vector 1111 to detect f SA1 without detecting either of the two faults it dominates, namely c SA0 and d SA0.

B. Error-Rate Test Generator for Fanout-Free Circuits

Now we will construct *ERTG-ff*, a test generator for error-rate testing for single stuck-at faults in fanout-free circuits with primitive gates. We then develop properties of test vectors generated by *ERTG-ff* to answer our central question.

We begin by rephrasing our central question in the following form: Given a fanout-free circuit with primitive gates and a target single stuck-at fault f_t , is it possible to generate a test vector for f_t such that $SATV(f) \supseteq SATV(f_t)$ for all single stuck-at faults f detected by the vector? Clearly, such a test generator will target a fault only if it is unacceptable.

Consider a generic fanout-free circuit shown in Fig 3, where every gate is a primitive gate. Consider a gate G with inputs x_1, x_2, \dots, x_p and output z . Let the target fault be $f_t = zSA\bar{\alpha}$. Excitation of the target fault necessitates application of value(s) at one or more inputs of G , namely x_1, x_2, \dots, x_p . The values applied at the input(s) of G must imply at its output, i.e., at the fault site z , a value $v(z) = \bar{\alpha}$ in a fault-free version of the circuit. Also, as described in Section IV.A and illustrated in Fig 1, the effect of the target fault must be propagated along the UPP with on-path gates G_1, G_2, \dots, G_{k-1} and on-path lines o_1, o_2, \dots, o_k . To propagate the fault-effect of f_t , it is necessary to apply at every side-input s_i of every on-path gate G_j , the gate's non-controlling value. In other words, any test vector for f_t must imply a value $v(s_i) = ncv(G_j)$, for every side input s_i of every on-path gate G_j .

Assignment of specific values at the fault site z and at every side input s_1, s_2, \dots, s_l of every on-path gate of the corresponding UPP implies specific values at on-path lines o_1, o_2, \dots, o_k . Let these values be $v(o_1), v(o_2), \dots, v(o_k)$. Note that all the above value assignments and implications are mandatory and must be satisfied by any test vector generated for f_t by any test generator.

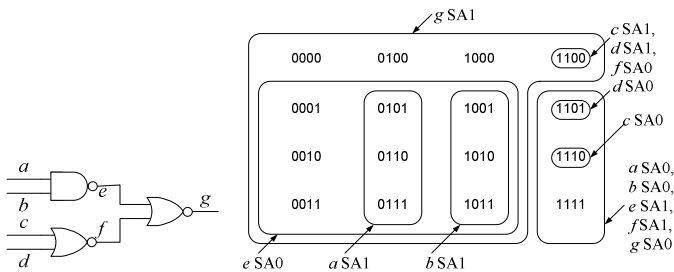


Fig 2. An example circuit and the relationships between the sets of all vectors that detect the single stuck-at faults.

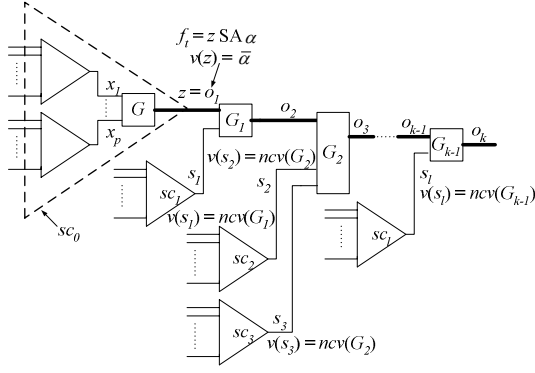


Fig 3. Generating a test vector for target fault f_i in a generic fanout-free circuit.

To obtain a test, the values at z and s_1, s_2, \dots, s_l must be *justified* (e.g., see [15]) by assigning values at the inputs of the fanout-free sub-circuits $sc_0, sc_1, sc_2, \dots, sc_l$ shown in Fig 3. In general, the desired value at the output of sub-circuit sc_i can be obtained using one of a number of combinations of values at the inputs of sc_i . Hence, we must develop a new justification procedure that identifies a combination of values at the inputs of sc_i that (i) implies at the output of sc_i the value desired for detection of the target fault f_i , and (ii) satisfies our other original objective, namely detects only faults f such that $SATV(f) \supseteq SATV(f_i)$. We develop such a justification procedure in Section IV.B.2.

1) Faults detected by any test vector for a target fault:

First, we deal with faults at on-path lines o_1, o_2, \dots, o_k and at off-path inputs s_1, s_2, \dots, s_l which are detected by any vector that detects the target fault. All results in this section hold for any vector that detects the target fault. In other words, these results hold for any test generation algorithm that can be used to generate a test vector for the target fault.

Lemma 1: Any vector that detects the target fault in a fanout-free circuit with primitive gates also detects a stuck-at fault at each on-path line along the unique propagation path from the fault site to the output. In particular, any vector that detects the target fault $f_i = z SA\alpha$ in the generic fanout-free circuit illustrated in Fig 3 also detects single stuck-at faults $o_1 SA\bar{v}(o_1), o_2 SA\bar{v}(o_2), \dots, o_k SA\bar{v}(o_k)$ at on-path lines.

The proof of this and every other result has been deleted due to lack of space but can be found in [7].

Lemma 2: The single stuck-at fault $o_i SA\bar{v}(o_i)$ at on-path line o_i detected by any vector that detects the target fault $f_i = z SA\alpha$ in the generic fanout-free circuit illustrated in Fig 3 is either equivalent to or dominates the target fault f_i . In other words, this fault satisfies the following conditions:

- (i) $SATV(o_i SA\bar{v}(o_i)) \supseteq SATV(f_i)$; and
- (ii) $R(o_i SA\bar{v}(o_i)) \geq R(f_i)$.

The above results collectively show that while it is impossible to detect the target fault f_i without also detecting

single stuck-at faults $o_1 SA\bar{v}(o_1), o_2 SA\bar{v}(o_2), \dots, o_k SA\bar{v}(o_k)$ at on-path lines o_1, o_2, \dots, o_k , since the target fault is unacceptable, each of these faults is also unacceptable.

Lemma 3: Any vector that detects the target fault f_i in a generic fanout-free circuit illustrated in Fig 3 also detects the fault $s_j SA\bar{v}(s_j)$ at a side input s_j of the on-path gate G_i , if and only if the single stuck-at faults at the side and on-path inputs of G_i , i.e., $s_j SA\bar{v}(s_j)$ and $o_i SA\bar{v}(o_i)$, are equivalent.

When we combine the result in Lemma 3 with that in Lemma 2, we see that fault $s_j SA\bar{v}(s_j)$ at a side input s_j of an on-path gate is detected if and only if the $s_j SA\bar{v}(s_j)$ is equivalent to or dominates the target fault f_i .

2) A new justification procedure:

Next we turn our attention to the justification procedure used to justify the values required at the fault site z and side-inputs s_1, s_2, \dots, s_l of on-path gates by applying specific combination of values at the inputs of the corresponding fanout-free sub-circuits $sc_0, sc_1, sc_2, \dots, sc_l$, respectively.

Let the objective of justification be to assign value $v(q)$ at line q , where q is the output of a primitive gate G with inputs q_1, q_2 , and so on. First consider the case where $v(q) = cr(G)$. In this case, the desired justification can be accomplished only by assigning the value $ncv(G)$ to all the inputs of G , i.e., by assigning $v(q_i) = ncv(G), \forall q_i$ that is an input to G . Next, consider the only other case, i.e., when $v(q) = cv(G)$. In this case, we can justify the desired value at the output of the gate by applying the gate's controlling value, i.e., $cv(G)$, to one or more of its inputs. If we apply the value $cv(G)$ to only one input of G , say q_j , then the single stuck-at fault $q_j SA\bar{ncv}(G)$ will be excited and its effect be propagated to q , the output of G . In contrast, if we apply the value $cv(G)$ to two or more inputs of G , then the effect of none of the stuck-at faults at any line in the transitive fanin of G will be propagated to the output q of G . In turn, this implies that none of the single stuck-at faults at the lines in the fan-in of G will be detected by the test vector generated for the target fault. Based on above observations, we use the procedure *Justify-ff-allcv* shown in Fig 4. To justify a desired value at the output of a fanout-free sub-circuit sc_i , this procedure is initially invoked with the line that is the output of sc_i and the desired value.

Justify-ff-allcv($q, v(q)$)

```
// q is output of gate G with controlling value cv(G)
// and controlled response cr(G)
if v(q) = cr(G)
  then for every input qi of G
    assign v(qi) = cv(G)
    if qi is not a primary input
      then Justify-ff-allcv(qi, v(qi))
else for every input qi of G
  assign v(qi) = ncv(G)
  if qi is not a primary input
    then Justify-ff-allcv(qi, v(qi))
```

Fig 4. The proposed justification procedure, Justify-ff-allcv.

ERTG-ff()

```
//  $G_i$  is a gate on the path from the fault site to the output
//  $s_k$  is a side input of the on-path gate  $G_i$ 
Select a target fault  $f_i, z SA\alpha$ , such that  $R(f_i) \geq T_{er}$ 
Justify-ff-allcv( $z, \bar{\alpha}$ )
for every on-path gate  $G_i$ 
  for every side input  $s_k$  of  $G_i$ 
    assign  $v(s_k) = ncv(G_i)$ 
    Justify-ff-allcv( $s_k, v(s_k)$ )
```

Fig 5. The proposed ERTG-ff procedure.

3) The proposed ERTG-ff and its properties:

The above justification procedure can be used in combination with the above concepts to obtain the proposed test generator for a single stuck-at fault in a fanout-free circuit with primitive gates. The proposed procedure is shown in Fig 5.

Note that we use ERTG-ff to explicitly generate a test for a fault only if the fault is unacceptable. Based on the description of ERTG-ff and the above discussion of the justification procedure Justify-ff-allcv, we now have the following results.

Lemma 4: Any vector generated by ERTG-ff for an unacceptable target fault f_i in a fanout-free circuit with primitive gates, detects a stuck-at fault $q SA\overline{v(q)}$ – where q is a line within the fanout-free sub-circuit sc_0 that drives the fault site z – if and only if the fault $q SA\overline{v(q)}$ is equivalent to f_i .

Lemma 5: Any vector generated by ERTG-ff for an unacceptable target fault f_i in a fanout-free circuit with primitive gates, detects a stuck-at fault $q SA\overline{v(q)}$ – where q is a line within the fanout-free sub-circuit sc_j that drives a side input s_j of an on-path gate G_i with on-path input o_i – if and only if the fault $q SA\overline{v(q)}$ is equivalent to the fault $o_i SA\overline{v(o_i)}$.

Finally, we can collectively use all the above procedures and lemmas to answer our central question.

Theorem 1: Any test vector generated by the procedure ERTG-ff for an unacceptable fault does not detect any acceptable fault.

The proposed test generator, ERTG-ff, provides a perfect solution for the problem of test generation for error-rate testing for single stuck-at faults in fanout-free circuits with primitive gates.

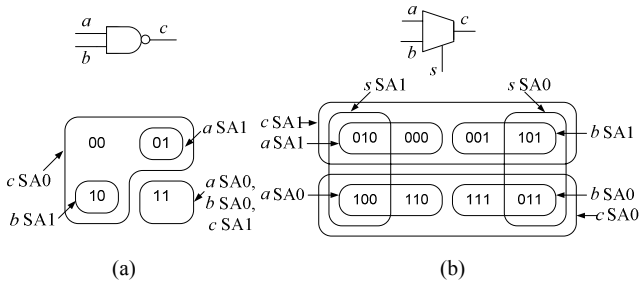


Fig 6. The sets of all test vectors for single faults associated with (a) a two-input NAND gate, and (b) a two-to-one multiplexer.

V. CHALLENGES POSED BY ARBITRARY CIRCUITS

In this section, we first analyze the challenges posed by some complex gates, even when they are used in fanout-free circuits. We then study circuits with non-reconvergent fanouts that only use primitive gates. Finally, we study the challenges posed by arbitrary combinational circuits which combine above challenges with new ones raised by the presence of reconvergent fanouts.

A. Some Complex Gates in Fanout-Free Circuits

The above concepts, procedures, and results can be easily extended to two commonly used families of complex gates, namely AOI and OAI. However, some commonly used gates do not have the properties that we have identified above. In this section, we will describe how some complex gates can pose new challenges to error-rate testing even in an otherwise fanout-free circuit.

Fig 6 shows the sets of all test vectors for single stuck-at faults located at inputs and the output of two different circuit elements. In the case of two-input NAND, $c SA0$ dominates $a SA1$ as well as $b SA1$. As described above, in a case where this gate is embedded within an arbitrary fanout-free circuit, $SATV(cSA0) \supset \{SATV(aSA1) \cup SATV(bSA1)\}$. This has two important consequences. First, the error rate for $c SA0$ is higher than the error rates for $a SA1$ and $b SA1$, independent of the probabilities of occurrence of values 00, 01, and 10 at the gate's inputs, a and b . Second, it is possible to use vectors that imply 00 at a and b to detect the dominating fault $c SA0$, which has the higher error rate, without detecting either of the two faults that it dominates, namely $a SA1$ and $b SA1$, which have lower error rates.

Next, consider the faults $c SA0$, $a SA0$, and $b SA0$ in the two-to-one multiplexer. Again, we see that $c SA0$ dominates each of the other two faults, and $SATV(cSA0) \supset SATV(aSA0)$ and $SATV(cSA0) \supset SATV(bSA0)$.

Again, the error rate for $c SA0$ is higher than the error rates for $a SA0$ and $b SA0$, independent of the probabilities of occurrence of values 100, 110, 111, and 011 at a , b , and s . However, $SATV(cSA0) = \{SATV(aSA0) \cup SATV(bSA0)\}$, and hence it is **not** possible to detect the dominating fault $c SA0$, which has the higher error rate, without detecting one of the two faults that it dominates, namely $a SA0$ and $b SA0$, which have lower error rates. Faults $s SA1$, $a SA0$, and $a SA1$ have a type of relationship that does not occur for primitive gates. The relative values of error rates of these three faults vary with the probabilities of occurrence of values 000, 010, 100, and 110 at the inputs of the multiplexer. Consider a case where the error rate for $s SA1$ is higher than those for $a SA0$ and $a SA1$. (This can occur, for example, if the values 010 and 100 occur at the lines a , b , and s with probabilities greater than those for values 000 and 110.) It is easy to see that in such cases, it is not possible to detect the fault with higher error rate, $s SA1$, without detecting at least one of the two faults with lower error rates. The reader is advised to check that similar complications arise in the case of a two-input XOR gate.

In general, when some complex gates are used in a fanout-free circuit, it may be impossible to detect some unacceptable faults without detecting at least one acceptable fault.

B. Circuits with Non-reconvergent Fanouts

Now consider circuits that have non-reconvergent fanouts. Fig 7(a) shows a three-gate circuit with a non-reconvergent fanout. Consider a scenario where every possible vector is applied with equal probability at the inputs of this three-gate circuit. In such a scenario, it is easy to compute the following error rates for the SA0 faults at the fanout stem e and its branches g and h : $R(eSA0) = 9/16$, $R(gSA0) = 6/16$, and $R(hSA0) = 6/16$. It is also easy to see why we cannot detect the stem fault e SA0, which has the higher error rate, without detecting the corresponding fault at one or more of the fanout branches, g SA0 or h SA0, each of which has a lower error rate. Note that this observation becomes important if the error rate threshold specified by the application and used to distinguish between unacceptable and acceptable faults is, $6/16 < T_{er} \leq 9/16$.

In general, even when a circuit with only non-convergent fanouts uses only primitive gates, it may be impossible to detect an unacceptable single stuck-at fault at a fanout stem without detecting at least one acceptable single stuck-at fault at one of its branches.

The number of acceptable faults detected by a complete set of test vectors for acceptable faults can be minimized. Consider a generic circuit with a non-reconvergent fanout shown in Fig 7(b) where a fault at the fanout stem, say a SA0 as well as some corresponding faults in the fanin of the stem a are unacceptable. Even if **any one of** the corresponding faults at the branches of this fanout, i.e., **either** b SA0 **or** c SA0, is acceptable, we can try to generate vectors that detect the unacceptable fault a SA0 and the corresponding unacceptable faults in the fanin of a , by propagating each of their fault effects via the fanout branch which has SA0 fault that is also unacceptable. Now consider a scenario where SA0 faults at both fanouts are acceptable. In such a case, we can generate vectors that detect unacceptable fault a SA0 and the corresponding unacceptable faults in the fanin of a , by propagating each of their fault effects via one particular fanout branch. Hence, even if each vector generated for an unacceptable fault detects a number of acceptable faults, the total number of acceptable faults detected by the overall set of test vectors can be small.

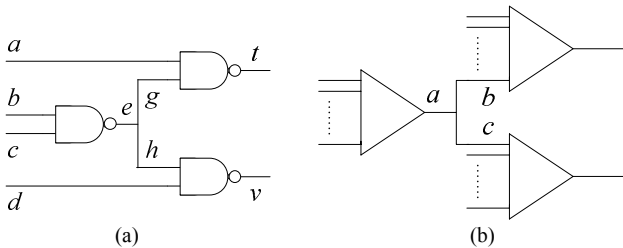


Fig 7. (a) An example circuit with non-reconvergent fanout, and (b) a generic circuit with non-reconvergent fanout.

C. Circuits with Reconvergent Fanout

In addition to the complications described above in an arbitrary combinational circuit the presence of reconvergent fanouts may make it (i) impossible to apply some combination of values at the inputs of a gate, or (ii) impossible to propagate the fault effect from the fault site to an output when a particular combination of values is applied to the inputs of the gate with the fault site to excite the fault.

VI. ARBITRARY CIRCUITS

An arbitrary combinational circuit may use complex gates, such as MUX and XOR, and may have non-reconvergent or reconvergent fanouts. We have answered our central question for fanout-free circuits with primitive gates by constructing a new test generator, ERTG-ff, that can generate a test vector for every unacceptable fault in any fanout-free circuit without detecting any acceptable fault. However, the challenges posed by the presence of certain complex gates and reconvergent and non-reconvergent fanouts lead us to the following conclusion: In an arbitrary combinational circuit, we cannot guarantee the detection of an unacceptable single stuck-at fault without detecting any acceptable fault.

This leads us to the second version of our central question raised in Section II: *What is the minimum number of acceptable faults that must be detected by a deterministic test set that detects every detectable unacceptable fault in an arbitrary combinational circuit?* In this section, we describe a test generation experiment that we carried out on benchmark circuits to provide a non-optimal and empirical answer to this question.

We assume that the error rate for every stuck-at fault in the circuit under test (CUT) is given and a threshold error rate is specified by the application. We use the above information to categorize every stuck-at fault in the circuit as acceptable or unacceptable. (Development of a systematic approach to estimate error rate for each fault is a subject of on-going research. For the following experiments we estimated error rate for each fault by simulating a large number of randomly-generated vectors.)

We target one unacceptable fault in the circuit at a time and use a classical ATPG to generate multiple test vectors for the target fault. We then simulate the circuit with each vector generated for the target fault to compute the number of acceptable faults covered by the vector. We select the vector for the target fault which detects the least number of acceptable faults and add the vector to the set of test vectors for the circuit. We then mark as detected every unacceptable fault detected by the selected vector. We generate and select a vector for every unmarked unacceptable fault in this manner. If we simply add the number of acceptable faults that are detected by each vector in our test set we obtain a number that is large. However we observed that total number of acceptable faults detected by the selected vectors is relatively small. This is because many vectors detect some of the same acceptable faults. Therefore, acceptable faults that are detected by a selected test vector are marked as already detected and

eliminated from simulations for vectors generated for subsequent faults. In other words, acceptable faults that are detected by a vector already added to the test set are not counted when we compute the number of acceptable faults that are detected by test vectors generated for subsequent faults.

This procedure was applied to several benchmark circuits. The results are summarized in Table I and show that it is indeed possible to generate test vectors that detect all unacceptable faults while detecting only a fraction of acceptable faults. Note that the above results are obtained using a computationally expensive ad-hoc approach where we generate multiple vectors for each unacceptable fault and use extensive fault simulation to select a vector. However, these results provide the motivation for developing a new test generation approach for error-rate testing with reasonable test generation complexity, high coverage of unacceptable faults, small number of detected acceptable faults, and minimum test size.

TABLE I
NUMBER OF ACCEPTABLE STUCK-AT FAULTS DETECTED BY A SET OF TEST VECTORS THAT DETECT ALL UNACCEPTABLE STUCK-AT FAULTS.

	T_{er}	Acceptable faults	Unacceptable faults	Acceptable faults detected
c880	0.0007	88 (5%)	1672 (95%)	5 (5.7%)
c432	0.01	62 (7.2%)	802 (92.8%)	20 (32.2%)
c880	0.01	440 (25%)	1320 (75%)	108 (24.7%)
c499	0.001	618 (61.9%)	380 (38.1%)	103 (16.7%)

VII. CONCLUSIONS AND ONGOING RESEARCH

In this paper we have developed concepts and algorithms and proven properties that provide the foundation for future research in error-rate testing. In particular, we have shown that it is possible to detect every unacceptable fault in a fanout-free circuit with primitive gates without detecting any acceptable fault. We have also described the key complications that arise in arbitrary circuits and explained why the above result may not hold for such circuits.

We then use an existing classical test generator to generate multiple vectors for each unacceptable fault in an arbitrary circuit and use extensive fault simulation to select a vector. This inelegant and computationally expensive approach served its intended purpose by providing empirical data which showed that it is indeed possible to generate test vectors that detect every unacceptable fault in an arbitrary circuit while detecting only a fraction of acceptable faults. In other words, it

showed that it is indeed possible to perform high quality error-rate testing while obtaining significant yield gains.

These results provide the motivation for our on-going research whose objective is to develop a new test generation approach for error rate which will have practical complexity, provide high coverage of unacceptable faults, detect even smaller number of acceptable faults, and minimize test set sizes so as to minimize test cost. We are currently generalizing the concepts, observations, properties, and algorithms developed in this paper to develop such a test generator.

REFERENCES

- [1] R.C. Leachman and C. N. Berglund, "Systematic Mechanisms Limited Yield (SMLY) Study", *International SEMATECH*, DOC #03034383A-ENG, March 2003.
- [2] M. A. Breuer and S. K. Gupta, "Intelligible Testing", *2nd IEEE Int'l Workshop on Microprocessor Test and Verification*, September, 1999.
- [3] M. A. Breuer, "Intelligible Test Techniques to Support Error-Tolerance", *proc. 13th Asian Test Symposium*, 2004.
- [4] M. A. Breuer, S. K. Gupta and T. M. Mak, "Defect and Error-Tolerance in the Presence of Massive Numbers of Defects", *IEEE Design and Test Magazine*, pp. 216-227, May-June 2004.
- [5] S. Shahidi and S. K. Gupta, "Estimating Error Rate during Self-Test via One's Counting", *proc. International Test Conference*, October 2006.
- [6] K. J. Lee, T. Hsieh and M. A. Breuer, "A Novel Test Methodology Based on Error-Rate to Support Error-Tolerance", *Proc. International Test Conference*, July 2005.
- [7] S. Shahidi and S. K. Gupta, "A Theory of Error-Rate Testing", Technical Report, Electrical Engineering Dept., University of Southern California, USC-CENG-2006-5, May 2006.
- [8] H. Chung and A. Ortega, "Analysis and Testing for Error Tolerant Motion Estimation", *Proc. 14th Defect and Fault Tolerance Conference*, Oct 2005.
- [9] I. S. Chong and A. Ortega, "Hardware Testing for Error Tolerant Multimedia Compression based on Linear Transforms", *Proc. 14th Defect and Fault Tolerance Conference*, Oct 2005.
- [10] J. Melzer, "Low Complexity Turbo-Like Codes", personal communication.
- [11] H. Zhu, "Error-Tolerance in Digital Speech Recording Systems", Master's Thesis, Electrical Engineering Dept., University of Southern California, 2006.
- [12] Z. Jiang and S.K. Gupta, "An ATPG for Threshold Testing: Obtaining Acceptable Yield in Future Processes", *Proc. International Test Conference*, July 2002.
- [13] E. J. McCluskey, and F. W. Clegg, "Fault Equivalence in Combinational Circuits", *IEEE Trans. on Computers*, 20 (11), 1971, pp. 1286-1293.
- [14] J. F. Poage and E. J. McCluskey, "Derivation of optimal Test Sequences for Sequential Machines", *Proc. Symposium on Switching theory and Logic Design*, 1964, pp. 121-132.
- [15] N. Jha and S. K. Gupta, *Testing of Digital Systems*, Cambridge University Press, 2003.
- [16] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, John Wiley and Sons, 1995.