compacted and trace-scheduled code in the shift and divide loops is readily explained. In each case in the sequential code the conditional jump is at the beginning of the loop, with an unconditional jump at the end back to the beginning. This structure is preserved in the trace-scheduled code. In the hand-coded version, the conditional branch is replicated at the end of the loop, thus avoiding the unconditional branch. We could incorporate such a specialized optimization into our machine compaction procedure. More ambitiously, we could develop a procedure to unroll (replicate) a loop, schedule the unrolled loop (possibly moving an operation from one iteration to another), and then reroll the loop (identifying repeating code segments) [4]. Such a procedure should be able to perform the optimization just cited.

The difference in the initialization segment of the floating add is more complex. At one point this code forks, with one path interchanging two registers, the other not doing so. The hand-compacted code *inserts* into the latter path two successive interchanges (an identity) and then moves the interchange now shared by both paths to before the fork. We do not see how this transformation could be readily incorporated into an automatic compacter.

We are encouraged that, except for this last instance, the trace scheduler performs or can be readily extended to perform as well as a skilled microprogrammer. We look forward to more extensive tests of trace scheduling and in particular to evaluations of the space-saving procedures suggested by Fisher.

ACKNOWLEDGMENT

Portions of the Setl code were borrowed from a program by R. Kenner.

REFERENCES

[1] J. Fisher, "Trace scheduling: A technique for global microcode compaction," *IEEE Trans. Comput.*, vol. C-30, pp. 478-490, July 1981.
[2] D. Landskov, S. Davidson, B. Shriver, and P. Mallett, "Local microcode compaction techniques," *ACM Comput. Surveys*, vol. 12, pp. 261-294, Sept. 1980.
[3] S. Davidson, D. Landskov, B. Shriver, and P. Mallett, "Some experiments in local microcode compaction for horizontal machines," *IEEE Trans. Comput.*, vol. C-30, pp. 460-477, July 1981.
[4] J. Fisher, D. Landskov, and B. Shriver, "Microcode compaction: Looking backward and looking forward," in *Proc. 1981 AFIPS Nat. Comput. Conf.* Arlington, VA: AFIPS Press, pp. 95-102.
[5] R. Grishman, "The Puma project: Computer design automation in the university," in *Proc. 1980 Annu. Conf. ACM*, pp. 490-497.
[6] R. B. K. Dewar, E. Schonberg, and J. T. Schwartz, "Higher level Programming: Introduction to the use of the set-theoretic programming language Setl," Courant Inst. Math. Sci. Comput. Sci. Dep., New York University, New York, NY, 1981.

Universal Tests for Detection of Input/Output Stuck-At and Bridging Faults

M. KARPOVSKY

***Abstract*—In this correspondence we present universal tests for detection of single and multiple stuck-at and bridging faults in combinational and sequential networks.**

Denote by $P_F(T, m, k)$ the fraction of all devices with m inputs and k outputs such that all faults from a set F are detected by the test T. We say that T is a universal test detecting all faults from F if $\lim_{m\to\infty} P_F(T, m, k) = 1$.

In this correspondence we consider stuck-at and bridging faults at input or output lines. For these faults we construct corresponding universal tests, estimate probabilities of fault detection, and present lower and upper bounds for minimum numbers of test patterns in universal tests. Asymptotic optimality of the suggested universal tests is proved. We also present practical examples of devices and tests which illustrate the usefulness of the estimates on minimum numbers of test patterns.

For the universal tests T proposed in this correspondence, probabilities $P_F(T, m, k)$ of fault detection converge to 1 very fast. This implies that these tests may be efficient either as a first step in a testing procedure or in the case when a broad spectrum of complex VLSI devices has to be tested.

***Index Terms*—Asymptotically optimal tests, fault detection, stuck-at and bridging faults, universal tests, upper and lower bounds for number of tests.**

Manuscript received April 5, 1983; revised May 10, 1983. This work was supported by the National Science Foundation under Grant MCS 8021262.

The author is with the Department of Computer Science, School of Advanced Technology, State University of New York at Binghamton, Binghamton, NY 13901.

I. INTRODUCTION

With the advent of VLSI technology, the cost of testing computer hardware is, in many cases, higher than the cost of development and manufacturing. It is well known that even in the case of single stuck-at faults, the problem of test generation is NP-hard [1], [2].

In this correspondence, we shall explore another approach to test generation, based on the idea of *universal* tests, detecting all faults from the given class for *almost all* devices. To define formally universal tests let us consider the set of all devices (with or without memory) with m input and k output binary lines. We suppose that each one of these devices has the same probability of being selected for testing. (If a device has a memory, we suppose that at the moment we apply each test pattern the device may be in each one of its internal states with the same probability.)

Let $T \subseteq \{0, 1\}^m$ and $P_F(T, m, k)$ is a probability of detection of all faults from the given class F by the test T. We shall say that T is the *universal* test, detecting faults from the class F for *almost all* devices, iff $\lim_{m\to\infty} P_F(T, m, k) = 1$ for any (not necessarily going to ∞) k. For all universal tests which we shall describe below, $P_F(T, m, k)$ goes to 1 very fast, and universal tests may be very efficient either as a first step in testing procedure or when we have to test a broad spectrum of devices.

We shall apply, in this correspondence, the universal testing approach to detection of faults at input/output pins since in many cases interconnections between chips are less reliable than chips themselves [3]-[6].

In view of this, we consider the following five classes of faults. 1) *Input stuck-at faults*, when each of the input lines may be stuck-at-0 or stuck-at-1. 2) *Output stuck-at faults*, when any number of output lines may be stuck, and each line may be stuck-at-0 or stuck-at-1. 3) *Input bridgings*, when any two input lines may be bridged. Two types of bridgings have been considered, namely, the AND-type and the OR-type. The AND and OR types of bridgings mean that two lines are short circuited to form AND and OR logical operations [4], [7]. Since for any given technology only one type of bridgings may appear in the device, we consider only AND-type bridgings. (Of course, all results may be easily reformulated also for the case of OR-type bridgings.) 4) *Output bridgings*. In this case we consider all bridgings between any number of output lines. 5) *Feedback bridgings*. These are bridgings between one input and one output line. As a result of these bridgings, a combinational network may behave as a sequential one; for example, it may oscillate or have an asynchronous behavior [3], [4]. In this correspondence we are not using oscillation and asynchronous behavior for detection of feedback bridgings.

For each one of these five classes of faults, and for the union of all five classes, we construct universal tests, estimate probabilities of error detection, and present lower $\underline{N}_F(m, k)$ and upper $\overline{N}_F(m, k)$ bounds for minimal numbers of test patterns. Upper bounds $\overline{N}_F(m, k)$ are constructive, and we describe universal tests such that $\overline{N}_F(m, k) = |T|$ where $|T|$ is the cardinality of T (a number of test patterns).

Definition: i) Test T is said to be *optimal* iff $\underline{N}_F(m, k) = \overline{N}_F(m, k)$. ii) Test T is *asymptotically optimal* iff

$$\lim_{m\to\infty} \frac{\overline{N}_F(m, k)}{\underline{N}_F(m, k)} = 1 \text{ or } \underline{N}_F(m, k) \sim \overline{N}_F(m, k) \text{ for any } k. \quad (1)$$

$(a(m) \lesssim b(m)$ iff $\lim_{m\to\infty} \frac{a(m)}{b(m)} \leq 1$; $a(m) \sim b(m)$ iff $a(m) \lesssim b(m)$ and $b(m) \lesssim a(m)$.)

All tests developed in this correspondence are optimal or asymptotically optimal.

It should be noted that sometimes results which are true for almost all cases have limited applications since in real life we are dealing with a tiny fraction of cases where these results are not valid.

As we shall see below, this is not the case when we are estimating minimal numbers of test patterns for almost all devices. For this case, we shall present simple examples of devices satisfying these estimations. For any device with m input lines and any test $T \subseteq \{0, 1\}^m$ with $N = |T|$ test patterns, we denote by (T) the $(N \times m)$ test matrix with rows of (T) corresponding to test patterns, and columns corresponding to input lines.

Let C be a binary error-correcting code, containing m codewords, with the Hamming distance d. We denote by T_C the test, such that columns of (T_C) are different codewords of C. In this case we shall say that $d(T_C) = d$. (We are considering codes without repetition, i.e., all columns in a generating matrix [8] for C are different.) Tests T_C will be used as universal tests for detection of bridging faults. We shall also use another type of tests which we denote by T_0^i, T_1^i, and T_{0-1}^i where $T_0^i = \{0^m, 10^{m-1}, 010^{m-2}, \cdots, 0^{i-1}10^{m-i}\}$, $T_1^i = \{1^m, 01^{m-1}, 101^{m-2}, \cdots, 1^{i-1}01^{m-i}\}$ $T_{0-1}^i = T_0^i \cup T_1^i$ $\left(\text{where } i \leq m, 0^j = \underbrace{0\cdots0}_{j}, 1^j = \underbrace{1\cdots1}_{j}\right)$. Tests T_0^i, T_1^i, and T_{0-1}^i will be used as the universal tests for the detection of stuck-at faults and feedback bridgings.

The following notations will be used throughout this correspondence.

$N_F(m, k)$	– minimal number of test patterns for detection of all faults from the class F in almost all devices with m input and k output lines;
$q_F(T, k)$	– probability of not detecting any fault from F by the test T;
$P_F(T, m, k)$	– probability of detecting all faults from F by the test T;
$\in(m)$	$= \frac{1}{\log_2 \log_2 m}$;
$\in(k)$	$= \frac{1}{\log_2 \log_2 k}$;
$\lceil a \rceil$	– smallest integer greater than or equal to a.

II. Detection of Input Stuck-at Faults

Denote:

$N_{IS}(m, k)$	– a minimal number of test patterns for detecting single input stuck-at faults;
$q_{IS}(T, k)$	– a probability of not detecting any given input stuck-at fault by the test T;
$a_0^{(j)}(T)(a_1^{(j)}(T))$	– a minimal number of zeros (ones) in the jth column of (T).

Theorem 1 (Lower Bound): For any k

$$N_{IS}(m, k) \geq 2\left\lceil \frac{\log_2 m}{k} \right\rceil. \quad (2)$$

Proof: For a minimal test T with $|T| = N$, denote

$$a^{(j)}(T) = \min(a_0^{(j)}(T), a_1^{(j)}(T)) \qquad (j = 1, \cdots, m). \quad (3)$$

Then for every $j \in \{1, \cdots, m\}$ there exists a stuck-at fault at the jth input line such that a probability of not detecting this fault by T is $2^{-a^{(j)}(T)k}$. Hence, $P_{IS}(T, m, k) \leq (1 - 2^{-A(T)k})^m$ where $A(T) = \max_j a^{(j)}(T)$. Thus, $P_{IS}(T, m, k) \to 1$ only if $A(T) \geq \left\lceil \frac{\log_2 m}{k} \right\rceil$. Since for every $j \in \{1, \cdots, m\}$ $a^{(j)}(T) \leq 0.5|T|$, we have $N_{IS}(m, k) = |T| \geq 2\left\lceil \frac{\log_2 m}{k} \right\rceil$.

Lemma 1: Any test T such that every column of (T) contains at least $a(T)$ zeros and at least $a(T)$ ones where $a(T) \geq \left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil$ detects all input stuck-at faults in almost all devices with m input and k output lines.

Proof: We have for any k $q_{IS}(T, k) \leq 2^{-a(T)k}$ and since there are $2m$ possible input stuck-at faults

$$\lim_{m\to\infty} P_{IS}(T, m, k) \geq \lim_{m\to\infty} (1 - 2^{-a(T)k})^{2m}$$
$$\geq \lim_{m\to\infty} (1 - m^{-(1+\in(m))})^{2m} = 1. \quad (4)$$

Theorem 2: i) For any k

$$2\left\lceil \frac{\log_2 m}{k} \right\rceil \leq N_{IS}(m, k) \leq 2\left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil. \quad (5)$$

ii) The test T_{0-1}^i with $i = \left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil - 1$ is the universal asymptotically optimal test for input stuck-at faults.

Proof: i) The lower bound follows from Theorem 1. ii) Take $T = T_{0-1}^i = \{0^m, 10^m, \cdots, 0^{i-1}10^{m-i}\} \cup \{1^m, 01^{m-1}, \cdots, 1^{i-1}01^{m-i}\}$ where $i = \left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil - 1$. Then every column of (T_{0-1}^i) contains $\left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil$ zeros and $\left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil$ ones and, by Lemma 1, T_{0-1}^i detects all input stuck-at faults. Thus, $N_{IS}(m, k) \leq |T_{0-1}^i| = 2(i + 1) = 2\left\lceil \frac{\log_2 m}{k}(1 + \in(m)) \right\rceil$.

Corollary 1: If $k - \log_2 m \to \infty$, then $N_{IS}(m, k) = 2$ (6)

Proof: Take $T = T_{0-1}^i$ with $i = 0$. Then from (4) with $a(T) = 1$,

$$\lim_{m\to\infty} P_{IS}(T_{0-1}^0, m, k) \geq \lim_{m\to\infty} (1 - 2^{-k})^{2m} = 1. \quad (7)$$

We note that, as it follows from (4), $P_{IS}(T_{0-1}^i, m, k)$ is converging to 1 very fast. For example, for $m = k = 32$ and $i = 0$, we have $T = T_{0-1}^0 = \{0^{32}, 1^{32}\}$, $|T_{0-1}^0| = 2$, $a(T) = 1$ and $P_{IS}(T_{0-1}^0, 32, 32) \geq 1 - 2^{-26}$.

Example 1: Let us consider an n bit combinational adder. For this case, $m = 2n$, $k = n + 1$, $k - \log_2 m \sim n - \log_2 n \to \infty$, and, in accordance with Corollary 1, $T = T_{0-1}^0 = \{0^{2n}, 1^{2n}\}$ detects all input stuck-at faults in n bit combinational adders.

We note also that a similar approach can be used for detection of input stuck-at faults of any given multiplicity l. In this case we have for the minimal number $N_{IS}^{(l)}(m, k)$ of test patterns

$$N_{IS}^{(l)}(m,k) \lesssim 2\left\lceil \frac{l\cdot \log_2 m}{k}\right\rceil. \tag{8}$$

III. Detection of Output Stuck-at Faults

Denote: $N_{OS}(m,k)$ — a minimal number of test patterns for detection of all output stuck-at faults of any multiplicity; $P_{OS}(T,m,k)$ — a probability of detection of all output stuck-at faults by the test T.

Theorem 3: If $k \to \infty$, then for any m,

$$\lceil \log_2 k\rceil \le N_{OS}(m,k) \le \lceil (1+\epsilon(k))\log_2 k\rceil \tag{9}$$

and any test T with $|T| \ge \lceil (1+\epsilon(k))\log_2 k\rceil$ is asymptotically optimal for output stuck-at faults.

Proof: For a given $T(|T| = N)$ denote by $(f(T))$ the matrix with elements $f(T)_{ij} = f_j(t^{(i)})$, where $f_j(t^{(i)})$, is the signal at the jth output if an input is $t^{(i)}$. Then, to detect all output stuck-at faults, it is necessary and sufficient that in every column of $(f(t))$ there is at least one "1" and one "0." Thus,

$$P_{OS}(T,m,k) = (1-2\,2^{-N})^k \qquad \text{for any } m. \tag{10}$$

Theorem 3 follows from (10) since

$$\lim_{m\to\infty}(1-2\cdot 2^{-N})^k = \begin{cases}1, & \text{if } N \ge \lceil 1+\epsilon(k))\log_2 k\rceil;\\ <1 & \text{if } N \le \lceil \log_2 k\rceil.\end{cases}$$

For example, if $k = 32$, then for any m and any T with $|T| = 16$ $P_{OS}(T,m,32) \ge 1 - 2^{10}$.

IV. Detection of Input Bridgings

Denote: $N_{IB}(m,k)$ — a minimal number of test patterns for detection of input bridgings; $q_{IB}(T,k)$ — a probability of not detecting any given input bridging by the test T; $P_{IB}(T,m,k)$ — a probability of detecting all input bridgings by the test T; $d(T)$ — a minimal Hamming distance between columns of (T). Let $n(S,d)$ be a length of the shortest binary code without repetition with S codewords and distance d. (The upper and lower bounds for $n(S,d)$ are well known [8, ch. 17, §5].)

Lemma 2: If $d(T)k - 2\log_2 m \to \infty$, (11)

then T detects all input bridgings and

$$N_{IB}(m,k) \le n(m, d(T)). \tag{12}$$

Proof: Since there are $\binom{m}{2}$ different input bridgings we have from (11)

$$\lim_{m\to\infty} P_{IB}(T,m,k) \le \lim_{m\to\infty}(1-2^{-d(T)k})^{\binom{m}{2}} = 1. \tag{13}$$

To estimate $n(m, d(T))$ we can use the Varshamov-Gilbert bound [8]. Thus, $N_{IB}(m,k)$ is asymptotically less or equal than the minimal N such that $N\left(1 - H_2\left(\frac{d(T)}{N}\right)\right) \gtrsim \log_2 m$

for $\frac{d(T)}{N} \lesssim 0.5$ where $H_2(\alpha) = -\alpha\log_2\alpha - (1-\alpha)\log_2(1-\alpha)$. (14)

Theorem 4: If $k \to \infty$, then $N_{IB}(m,k) \sim \log_2 m$. (15)

Proof: i) If T detects all input bridgings, then all columns of (T) are different

and

$$N_{IB}(m,k) \ge \lceil \log_2 m\rceil. \tag{16}$$

ii) Take $d(T) = \left\lceil (2+\epsilon(m))\frac{\log_2 m}{k}\right\rceil$. Then (11) is satisfied, and there exists a code C with length $N \sim \log_2 m$ and $|C| \ge m$ since in this case $\frac{d(T)}{N} \to 0$ for $k \to \infty$. [8]. Then by Lemma 2, $T = T_C$ detects all input bridgings and $N_{IB}(m,k) \le |T_C| \sim \log_2 m$.

Thus, for input bridgings as $k \to \infty$ any T such that $d(T) \ge \left\lceil (2+\epsilon(m))\frac{\log_2 m}{k}\right\rceil$ and $|T| \sim \log_2 m$ is asymptotically optimal. We can take $T = T_C$ where C is a code meeting the Varshamov-Gilbert bound [8]. If $k \ge \epsilon(m)\log_2 m$, then we can choose as C a BCH code [8] correcting $\lceil \log_2\log_2 m\rceil$ errors. In this case $d(T_C)k - 2\cdot\log_2 m \ge (2\log_2\log_2 m + 1)\epsilon(m)\log_2 m - 2\cdot\log_2 m = \epsilon(m)\log_2 m \to \infty$, and (11) is satisfied.

Corollary 2: If $k \ge (2+\epsilon(m))\log_2 m$, then $N_{IB}(m,k) = \lceil \log_2 m\rceil$, and the test T_1, such that all columns of (T_1) are different, is optimal.

Proof: For the test T_1 we have $|T_1| = \lceil \log_2 m\rceil$, $d(T_1) = 1$, $d(T_1)k - 2\cdot\log_2 m \ge \epsilon(m)\log_2 m \to \infty$, and Corollary 2 follows from Lemma 2 and (16).

Example 2: For an n bit multiplier we have $m = k = 2n$ and, by Corollary 2, T_1 detects all input bridgings. The minimal number of test patterns to detect input bridgings in n bit multipliers is $\lceil \log_2 m\rceil = 1 + \lceil \log_2 n\rceil$.

If m, k, and the lower bound P_{IB} for $P_{IB}(m,k)$ ("fault coverage") are given, then to construct the best test T_{IB}, we first compute a minimal d such that

$$(1-2^{-dk})^{\binom{m}{2}} \ge P_{IB}, \tag{17}$$

then construct the shortest C such that $|C| \ge m$ and the distance of C is at least d. For T_{IB}, columns of (T_{IB}) are different codewords of C. We note also, that $N_{IB} = |T_{IB}|$ increases when m increases or k decreases for any given P_{IB}.

V. Detection of Output Bridgings

Denote: $N_{OB}(m,k)$ — a minimal number of test patterns for the detection of all output bridgings; $P_{OB}(T,m,k)$ — a probability of detection of all output bridgings by the test T.

Theorem 5: For $k \to \infty$ and any m $\lceil 2\log_2 k\rceil \le N_{OB}(m,k) \le \lceil (2+\epsilon(k))\log_2 k\rceil$, (18)

and any test T such that $|T| = \lceil (2+\epsilon(k))\log_2 k\rceil$ is asymptotically optimal for the detection of output bridgings.

Proof: To detect all output bridgings it is necessary and sufficient that all columns in $(f(T))$ are different $(f(T)_{ij} = f_j(t^{(i)}))$. Thus, $|T| = N \ge \lceil \log_2 k\rceil$ and for any m

$$P_{OB}(T,m,k) = \prod_{r=0}^{k-1}(1 - r2^{-N}). \tag{19}$$

Theorem 5 follows now from (19) since $\lim_{k\to\infty}\prod_{r=0}^{k-1}(1-r2^{-N}) = \begin{cases}1, & \text{if } N \ge (2+\epsilon(k))\log_2 k;\\ <1, & \text{if } N \le \lceil 2\log_2 k\rceil.\end{cases}$

For example, for $m = k = 32$ and $N = 20$ we have from (19) for any T with $|T| = 20$

$$P_{OB}(T,32,32) = \prod_{r=0}^{31}(1-r2^{-20}) \ge 1 - 2^{-11}.$$

Corollary 3: For $k \to \infty$ and any m

$$N_{OB}(m,k) \sim 2\log_2 k. \tag{20}$$

Proof: Corollary 3 follows immediately from Theorem 5.

VI. Detection of Feedback Bridgings

The AND-bridging between input x_i and output y_j is detected by a test pattern $t = (t_1,\cdots,t_i,\cdots,t_m)$ iff $t_i = 0$ and $f_j(t_1,\cdots,t_{i-1},0,t_{i+1},\cdots,t_m) = 1$.

Denote: $N_{FB}(m,k)$ — a minimal number of test patterns for detecting feedback bridgings; $q_{FB}(T, k)$ – a probability that any given feedback bridging is not detected by the test T; $P_{FB}(T, m, k)$ – a probability of detecting all feedback bridgings by T; $a_0(T)$ and $a_1(T)$ – minimal numbers of zeros and ones in columns of (T).

Theorem 6 (Lower Bound): For any m and $k \to \infty$

$$N_{FB}(m, k) \geq \lceil \log_2 k \rceil. \quad (21)$$

Proof: If $f_j(t) = t_i$, then the bridging between x_i and y_j is not detected by t. If T is a minimal test with $|T| = N$, then for any k $q_{FB}(T, k) \geq 2^{-N}$ and for any m

$$\lim_{k\to\infty} P_{FB}(T, m, k) \leq \lim_{k\to\infty} (1 - 2^{-N})^k. \quad (22)$$

Thus, $P_{FB}(T, m, k) \to 1$ only if $N = N_{FB}(m, k) > \log_2 k$.

Theorem 7: If $m \geq \lceil (1 + \epsilon(k)) \log_2 k \rceil$ and $k \to \infty$, then (23)

$$\lceil \log_2 k \rceil \leq N_{FB}(m, k) \leq \lceil (1 + \epsilon(k)) \log_2 k \rceil. \quad (24)$$

Proof: Take $T = T_0^i$ with $i = \lceil (1 + \epsilon(k)) \log_2 k \rceil - 1$. Then every column and row of (T_0^i) contains at most one 1. If there exists $t, \tau \in T_0^i$ such that $f_j(t) = f_j(\tau) = 1$, then feedback bridgings between any input and y_j are detected by T_0^i. The probability of this is at least $1 - (N + 1)2^{-N}$. Thus, $P_{FB}(T_0^i, m, k) \geq (1 - (N + 1)2^{-N})^k$ and

$$\lim_{k\to\infty} P_{FB}(T_0^i, m, k) \geq \lim_{k\to\infty} (1 - (N + 1)2^{-N})^k$$
$$\geq \lim_{k\to\infty} (1 - (1 + \epsilon(k)) (\log_2 k + 1)k^{-(1+\epsilon(k))})^k = 1. \quad (25)$$

It follows from Theorem 7, that for $k \to \infty$ and $m \geq \lceil (1 + \epsilon(k)) \log_2 k \rceil$ the test T_0^i with $i = \lceil (1 + \epsilon(k)) \log_2 k \rceil - 1$ is asymptotically optimal.

Let us estimate the probability of detection of feedback bridgings for $m = k = 32$. Take $T = T_0^{20}$, then $N = 21$ and from (25) $P_{FB}(T_0^{20}, 32, 32) \geq 1 - 2^{-11}$.

We note that Theorems 6 and 7 remain valid also for OR-type bridgings. To prove this we only have to replace T_0^i by T_1^i.

VII. Detection of All Input/Output Stuck-at and Bridging Faults

In this section we estimate minimal numbers $N(m, k)$ of test patterns for the detection of all input/output, stuck-at, and bridging faults and construct universal tests for these faults. As before we suppose that $k \to \infty$ and (23) is satisfied.

Theorem 8:

$$\max(\lceil \log_2 m \rceil, \lceil 2 \log_2 k \rceil) \leq N(m, k) \leq \lceil (1 + \epsilon(k)) \log_2 k \rceil$$
$$+ \max\left(n\left(m + 2, \left\lceil 2 (1 + \epsilon(m)) \frac{\log_2 m}{k} \right\rceil\right), \lceil \log_2 k \rceil\right) + 1. \quad (26)$$

Proof: The lower bound follows from (16) and (20). Let C be a linear code with the distance $d = \left\lceil 2(1 + \epsilon(m)) \frac{\log_2 m}{k} \right\rceil$, length at least $\lceil \log_2 k \rceil$ and $|C| \geq m + 2$. Then C contains at most one vector v with a number of zeros less than $\left\lceil (1 + \epsilon(m)) \frac{\log_2 m}{k} \right\rceil$. Let columns of (T_C) be codewords of C which are not all zeros and v. Then

$$|T_C| = \max\left(n\left(m + 2, \left\lceil 2(1 + \epsilon(m)) \frac{\log_2 m}{k} \right\rceil\right), \lceil \log_2 k \rceil\right), \text{ and } d(T_C) = d. \quad (27)$$

Take $T = T_0^i T_C$ where $i = \lceil (1 + \epsilon(k)) \log_2 k \rceil$, then $a_0(T) \geq \left\lceil \frac{\log_2 m}{k} (1 + \epsilon(m)) \right\rceil$; $a_1(T) \geq d(T_C) \geq \left\lceil \frac{\log_2 m}{k} (1 + \epsilon(m)) \right\rceil$ and, by Lemma 1, T detects all input stuck-at faults. Since $|T| > i = \lceil (1 + \epsilon(k)) \log_2 k \rceil$, by Theorem 3, T also detects all output stuck-at faults. Since $d(T_C)k - 2 \cdot \log_2 m \to \infty$, $|T| - 2 \cdot \log_2 k \to \infty$ and $a_0(T) \geq \lceil (1 + \epsilon(k)) \log_2 k \rceil$, all input, output, and feedback bridgings are also detected by T. From (27) we have $N(m, k) \leq |T| = |T_C| + i + 1 = \max\left(n(m + 2, \left\lceil 2(1 + \epsilon(m)) \frac{\log_2 k}{k} \right\rceil, \lceil \log_2 k \rceil\right) + \lceil (1 + \epsilon(k)) \log_2 k \rceil + 1$.

Corollary 4:

$$N(m, k) \sim \begin{cases} \log_2 m, & \text{if } \frac{k}{m} \to 0; \\ 2 \log_2 k, & \text{otherwise.} \end{cases} \quad (28)$$

Proof: Corollary 4 follows from Theorem 8 in view of [8]

$$n\left(m + 2, \left\lceil 2(1 + \epsilon(m)) \frac{\log_2 m}{k} \right\rceil\right) \sim \log_2 m.$$

Thus, $T = T_0^i \cup T_C$ with $i = \lceil (1 + \epsilon(k)) \log_2 k \rceil$ is asymptotically optimal for the detection of all input/output stuck-at and bridging faults. In the case $\lim_{m\to\infty} \frac{k}{m} \neq 0$, we can use the following result.

Corollary 5: If $k \geq \lceil (2 + \epsilon(m)) \log_2 m \rceil$, then $T = T_0^i \cup T_1$, where $i = \lceil (1 + \epsilon(k)) \log_2 k \rceil$, all columns of (T_1) are different and not equal to 0, is asymptotically optimal for the detection of all input/output stuck-at and bridging faults. For this test $|T| = \lceil (1 + \epsilon(k)) \log_2 k \rceil + \lceil \log_2 (m + 1) \rceil + 1$.

Proof: We can use the same proof as the proof of Theorem 8, taking into account that in this case $d(T) \geq d(T_1) = 1$ and $a_1(T) = 1$, $a_0(T) \geq i = \lceil (1 + \epsilon(k)) \log_2 k \rceil$.

We note that for such important (from a practical point of view) devices as shifters, counters with the parallel load, adders, subtractors, multipliers, etc., we have $k = rm$ $(0.5 \lesssim r \lesssim 1)$, and minimal numbers of test patterns for the detection of input/output stuck-at and bridging faults are between $\log_2 k$ and $2 \log_2 k$ [3], [4], which illustrate the practical usefulness of Theorem 8 and Corollaries 4 and 5.

Numbers of test patterns for detection of input/output faults with a probability at least $1 - 2^{-10}$ are given in Table I.

TABLE I
MINIMAL NUMBERS OF TEST PATTERNS FOR DETECTION OF ALL INPUT/OUTPUT STUCK-AT AND BRIDGING FAULTS WITH A PROBABILITY AT LEAST $1 - 2^{-10}$. (FOR COMPUTATIONS IN TABLE I THE TABLES OF THE BEST CODES FROM [8, APPENDIX A, §1] HAVE BEEN USED.)

$k \backslash m$	30	62	126	254	510
1	54	58	65	73	75
2	37	42	43	50	52
4	30	32	33	37	38
8	28	29	30	31	32
16	26	27	28	29	30
32	26	27	28	29	30
64	27	28	29	30	31
128	29	30	31	32	33
256	30	31	32	33	34
512	31	32	33	34	35

Acknowledgment

The author wishes to thank Dr. L. Levitin for many useful comments.

References

[1] O. H. Ibarra and S. Sahni, "Polynomially complete fault detection problems," *IEEE Trans. Comput.*, vol. C-24, pp. 242–249, Mar. 1975.

[2] S. Sahni and A. Bhatt, "The complexity of design automation problems," in *Proc. 17th Design Automation Conf.*, Minneapolis, MN, May 1980, pp. 402–411.

[3] M. Karpovsky and S. Su, "Detection and location of input and feedback bridging faults among input and output lines," *IEEE Trans. Comput.*, vol. C-29, pp. 523–527, June 1980.

[4] ——, "Detection of bridging and stuck-at faults at input output pins of standard computer components," in *Proc. 17th Design Automation Conf.*, Minneapolis, MN, May 1980, pp. 494–506.
[5] C. D. Weiss, "Bounds on the length of terminal stuck-fault tests," *IEEE Trans. Comput.*, vol. C-21, pp. 305–308, 1972.
[6] J. G. Kuhl and S. M. Reddy, "On detection of terminal stuck-faults," *IEEE Trans. Comput.*, vol. C-27, pp. 467–468, May 1978.
[7] K. L. Kodandapani and D. K. Pradhan, "Undetectability of bridging faults, and validity of stuck-faults test sets," *IEEE Trans. Comput.*, vol. C-28, pp. 55–59, Jan. 1980.
[8] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North-Holland, 1977.

Good Controllability and Observability Do Not Guarantee Good Testability

JACOB SAVIR

***Abstract*—In this paper we show that good controllability and observability do not guarantee good testability. In fact, one can easily find examples of faults that are difficult or impossible to detect, although both the controllability and observability figures are good.**

***Index Terms*—Controllability, deterministic testing, observability, random testing, testability.**

Introduction

The problem of analyzing the testability of a digital circuit has long been recognized to be an important one. With the levels of integration existing today, the cost of testing and diagnosis have become so large, that they are a significant part of the cost of the product. In order to reduce this cost it is crucial to have highly testable circuits. Since test generation and fault simulation consume a lot of computer time in present day densities, it is worthwhile to be able to predict whether or not the testing task is going to be easy.

A few testability measures and programs that implement them have been reported to date [1]–[5]. The limitations of these measures are:

1) The controllability, observability, and testability measures are not an accurate measure to the "ease of testing."

2) They fail to report testability problems in the presence of reconverging fanout.

3) The testability measures are defined such that "good controllability and observability figures usually imply good testability," which is not true in many cases.

4) Because the measures are not a true reflection of the ease of testing, they may guide the test designer to introduce hardware real estate (to enhance testability) in the wrong place.

In this paper we elaborate on these issues. The paper should not be regarded as a "new testability measure," but rather as an attempt to point out the limitations of the existing methods, and the kind of emphasis necessary from the future ones to come.

The discussion is restricted to combinational circuits and stuck-at-faults.

I. Definitions and Properties

Let C be a combinational circuit with n inputs, $x_1, x_2, \cdots, x_n$, and m outputs, $F_1, F_2, \cdots, F_m$. Let $\bar{x} = (x_1, x_2, \cdots, x_n)$. Let $g(\bar{x})$ be a line in the circuit. We denote by $g/0$ ($g/1$) the fault g stuck-at-zero (g stuck-at-one).

Manuscript received June 5, 1982; revised November 10, 1982.
The author is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

The controllability of a line in the circuit is a measure of how easy it is to set the line to a given value. Similarly, the observability of a line is a measure of how easy it is to observe its value. We define the controllability and observability of a fault in the following way.

Definition 1: The *controllability of a fault g/i*, $i \in \{0, 1\}$ is the fraction of input vectors that will set the value of that line to $\bar{i}$. In other words, the controllability of a fault g/i is the probability that an input picked at random will set the value of line g to $\bar{i}$.

Definition 2: The *observability of a fault g/i*, $i \in \{0, 1\}$, is the fraction of input vectors that will propagate the effect of this fault to a primary output. In other words, the observability of a fault is the probability that an input picked at random will propagate the effect of this fault to a primary output.

The input vectors that detect the fault $g/0$ can be obtained by solving the Boolean equation

$$g(\bar{x}) \sum_{j=1}^{m} \frac{\partial F_j}{\partial g} = 1, \tag{1}$$

and the input vectors that detect the fault $g/1$ can be obtained by solving the equation

$$\bar{g}(\bar{x}) \sum_{j=1}^{m} \frac{\partial F_j}{\partial g} = 1 \tag{2}$$

where the summation symbol means the Boolean sum (OR operation). Thus, according to definitions 1 and 2, the controllability of the fault $g/0$, $c(g/0)$, is the fraction of input combinations that yield $g(\bar{x}) = 1$, and the controllability of the fault $g/1$, $c(g/1)$, is the fraction of input vectors that yield $\bar{g}(\bar{x}) = 1$. Similarly, the observability o of either fault ($g/0$, or $g/1$) is the fraction of the input vectors that yield

$$\sum_{j=1}^{m} \frac{\partial F_j}{\partial g} = 1. \tag{3}$$

It is worth while to note that the notion of the *syndrome* [6] of a function F, denoted by $S(F)$, is exactly the fraction of the input vectors that yield $F = 1$. Thus, we can take advantage of syndrome relations to compute controllability and observability figures. In particular, the following relations hold:

$$c(g/0) = S(g(\bar{x})), \tag{4}$$

$$c(g/1) = S(\bar{g}(\bar{x})), \tag{5}$$

and

$$o(g/0) = o(g/1) = S\left(\sum_{j=1}^{m} \frac{\partial F_j}{\partial g}\right). \tag{6}$$

Definition 3: The testability of a fault g/i, $i \in \{0, 1\}$ is the fraction of the input vectors that detect the fault.

In other words, the testability of a fault is the probability that an input picked at random will detect the fault.

According to Definition 3 and the notion of the syndrome, we can relate the testabilities $t(g/0)$ and $t(g/1)$ of the faults $g/0$ and $g/1$ to the following syndrome relations:

$$t(g/0) = S\left(g(\bar{x}) \sum_{j=1}^{m} \frac{\partial F_j}{\partial g}\right), \tag{7}$$

$$t(g/1) = S\left(\bar{g}(\bar{x}) \sum_{j=1}^{m} \frac{\partial F_j}{\partial g}\right). \tag{8}$$

The definitions listed above, and the properties of the syndrome, further imply the following relations between controllability, observability, and testability of a fault:

Property 1:

$$0 \le c(g/i), o(g/i) \le 1, \text{ for } i \in \{0, 1\} \tag{9}$$