35

UNIVERSAL TESTING OF COMPUTER HARDWARE[1]

M. G. Karpovsky
L. B. Levitin
College of Engineering
Boston University
Boston, MA, USA

[1] This work was supported by the National Science Foundation under Grant DCR-8317763

ISBN 0-12-400060-6

I. INTRODUCTION

The vigorous modern development of VLSI technology poses new difficult problems in the testing of digital systems hardware. Micro-miniaturization of components and the increase of packing density lead to inevitable faults, so that at some stages of production the yield of faulty chips reaches 20% - 30% and more. Test generation and application become more and more time- and money- consuming and threaten to turn into a real bottleneck of the computer industry. The time and the cost of testing increases tremendously (in general, exponentially) with the complexity of devices to be tested. The cost of testing is already, in many cases, higher than the cost of development and manufacturing. 'The majority of direct labor time required to build an Apple III is straight test time..... Testing requires 35% of the direct labor time involved in building a system.... If the test and the rework time is added together, 72% of the process time is accounted for' [1].

The following three approaches have been traditionally used for testing, namely: gate-level testing, functional testing and random testing. For the gate-level testing the input data for test generation consist of a gate-level description of a device under test and a gate-level description of a class of possible faults. This approach has been proved to be very efficient for SSI and MSI circuits [2, 3, 4]. The classical D-algorithm [3] has been widely used for gate-level testing for many years. With the transition to VLSI technology the gate-level description of a device under test becomes too complicated and in many cases (in particular, for the user) is not available at all. Even if the gate-level description is available to a test designer, the cost of test generation begins to be prohibitively high for VLSI devices [5]. It is well known that the problem of optimal test generation for VLSI circuits is NP-hard [6, 7] even in the case of single stuck-at faults. It implies that the cost of test generation increases exponentially with the number of gates, so that the development of a specific test for every device becomes impractical when a broad spectrum of complex devices has to be tested.

These considerations stimulated the development of functional testing approach, especially for microprocessors testing. In the case of functional testing, the input data for testing is represented by a functional description of a device under test and of a class of faults [8 - 15]. In [9, 10] the functional testing approach has been used for testing of microprocessors in a user's environment. This procedure was based on a general graph-theoretical model of a microprocessor at the register-transfer level. The functional fault model was developed using the register-transfer level description and the instruction set. Binary decision diagrams have been used for functional testing in [11], where this approach has been applied to testing of ALU.

For functional testing the cost of test generation for VLSI devices is still very high, especially when a broad spectrum of devices has to be tested. On the other hand, it is very difficult, in general, to estimate fault coverage in the case of functional testing. For random testing test patterns are generated randomly [16 - 25]. For generation of pseudorandom test patterns, linear feedback shift registers have been used [26]. Random testing has been used both at the gate level and at the functional level. In the latter case a random sequence of instructions has been used for testing (see, e.g., [24]). The major problem in random testing is to estimate the test length and the probability of fault detection [23, 24]. For random testing the cost of test generation is not high but a number of test patterns (testing time) may be very high for VLSI devices.

There is one more approach for testing, when one does not try to develop a test for a specific device but constructs a deterministic standard test which can be used for a large set of devices. One example of this approach is presented in [27] where exhaustive testing and simple data compression scheme have been used for testing. In [28-34] standard tests have been developed for combinational devices such that every output depends only on a small subset of inputs.

In this chapter we describe another approach for testing which we call 'universal testing'. This is a probabilistic approach which is somewhat similar to standard tests and aimed to fill the gap between the functional and random testing, combining the advantages of both of them. For universal testing [35 - 38] the input data for test generation consists of parameters of a device under test (e.g., numbers of input and output lines, numbers of flip-flops, etc) and description of a class of possible faults (e.g., stuck-at faults of a given multiplicity). Universal tests may be very efficient for a testing of a broad spectrum of devices or as a first step in a testing procedure.

Section II of this chapter contains a description of devices under consideration, definitions and notations as well as general results on probabilities of detection (or identification) of all faults from a given class. These results are applied to detection of faults at input/output (terminal) lines of a device (so called 'terminal' faults). Functional testing approach for these faults has been considered in [39, 40]. Terminal faults in both combinational and sequential circuits as well as some internal faults and faults in memory are also considered in Section II. As a fault model we choose single and multiple stuck-at and bridging (short-circuit) faults when a short-circuit results in a wired AND or wired OR logic between wires which are bridged (see, e.g. [41, 42]). (With the increase of packing density in VLSI devices bridging faults begin to be a very important class of faults. A number of papers have been published on detection of bridging faults or combined stuck-at and bridging faults [39 - 44].)

The comparison of gate-level testing, functional testing, random testing and universal testing is given in Section III. Possible directions for future research are outlined in Section IV.

In this chapter we do not consider design for testability, data compression and self-test aspects of the testing problem. The reader is referred to Chapter 7 and 8 of this book for data compression techniques and to Chapter 9 for self-test approaches, as well as to the references given in the corresponding chapters.

II. BASIC IDEAS OF UNIVERSAL TESTING AND APPLICATIONS TO COMBINATIONAL CIRCUITS

A. General Remarks

As it has been discussed above, the problem of test generation becomes intractable for complex circuits (with large numbers of inputs, gates and connecting lines), if one tries to find an optimal specific test for any given circuit. Even to construct a test which is not optimal, but provides a good fault coverage (say, > 95%) is quite difficult. It results usually in an excessively large size of a test (in some cases the tests include tens of thousands of test patterns) and/or in adding redundant hardware to the circuits in order to provide better and simpler testability (so called design for testability [45,46]).

Fortunately, the same reasons that make the problem of generation of individual tests for various VLSI circuits so difficult - namely, the increasing complexity and large diversity of the circuits - open a way to a different approach to test generation. This approach has been recently developed in [35 - 38] and is called <u>universal testing</u>. To explain the basic idea of this approach we need to turn from the deterministic concept of testing to the probabilistic one. It is somewhat similar in ideology to the transition from mechanics to statistical mechanics. The major breakthrough in communication engineering owing to Shannon's information theory was based on very similar ideas [47 - 50]. The same philosophy was successfully applied in some other areas, such as collective behavior of automata [51 - 52], neural networks [53], etc.

This way of reasoning brings us to the idea of <u>universal tests</u> the tests that are able to detect <u>all the faults</u> of a given class in <u>almost all circuits</u> of a given ensemble of circuits. The fraction of the circuits in which a universal test detects all the faults of a given class approaches one when the circuit complexity increases.

More rigorously, the concept of universal testing can be defined in the following way. Let $C_i = \{c_{ij}\}$ be a finite set of circuits c_{ij} of a given complexity, the complexity increasing with i. Suppose that a circuit c_{ij} can be chosen randomly from the set C_i with probability p_{ij} ($\sum_j p_{ij} = 1$). Denote by F_i a set of faults of given type which can occur in any $c_{ij} \in C_i$. Consider a test T_i which is applicable to any c_{ij}. Denote by

$P(T_i, F_i, C_i)$ the probability that the test T_i detects all the faults from F_i in a circuit c_{ij} randomly chosen from C_i with probability p_{ij}.

Definition 1. A sequence of tests (T_i) is called <u>universal</u> for detection of all faults of type F if

$$\lim_{i\to\infty} P(T_i,F_i,C_i)= 1 \qquad . \qquad (1)$$

To avoid misunderstanding, we would like to stress that the probability P is the <u>fraction of the circuits</u> for which the universal test provides <u>complete (100%) fault coverage</u> (and <u>not</u> the measure of fault coverage for a given circuit).

One should distinguish between our use of the term 'universal tests' and the way this term is used in [54, 55]. Our approach is essentially probabilistic, and the universality is understood in the asymptotic meaning, as expressed in Def. 1. No use of additional hardware and no special design is assumed to improve testability of the circuits. The authors of [54,55] suggest tests which can detect some classes of faults in <u>all</u> PLA with given numbers of inputs and product terms (in a deterministic way), but they can do it only by the expense of <u>adding extra logic</u> and designing the PLAs in a special way. Thus, the approach adopted in [54, 55], belongs, in essence, to the design for testability area, and differs in principle from our approach.

B. Definitions and Notations.

An arbitrary digital binary circuit with memory may be represented by a block-diagram of Fig. 1.

Here m is the number of input lines, k is the number of output

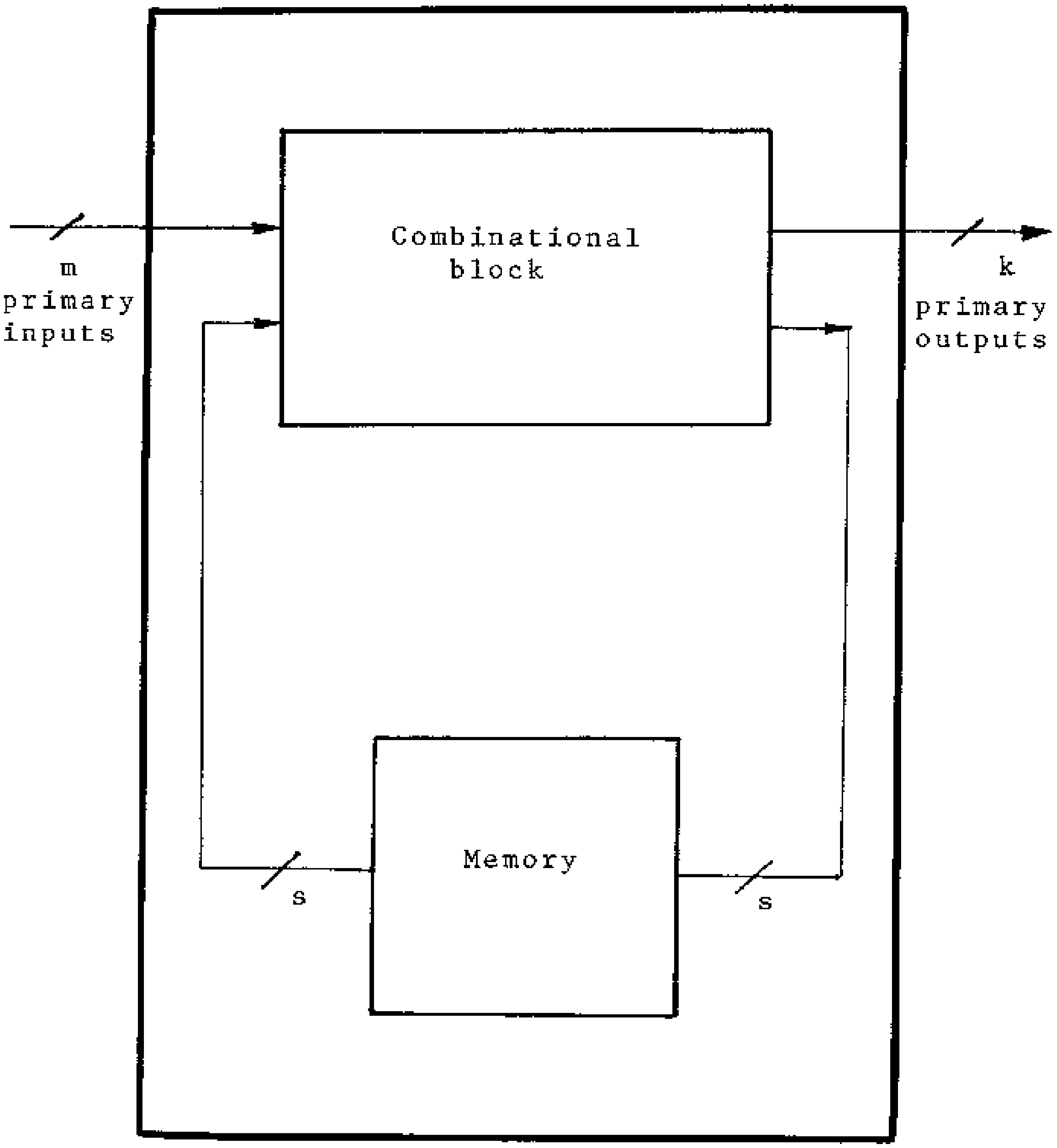

Fig. 1. Block-diagram of a device with memory.

lines, and s is the number of feedback lines (the number of binary memory cells). Such a device will be referred to as an (m,k,s) - circuit, so that an (m,k,0) - circuit is a combinational circuit.

Denote also:

$x = (x_1,\dots,x_m)$ is an input binary vector,
$y = (y_1,\dots,y_k)$ is an output binary vector,
$z = (z_1,\dots,z_s)$ is a feedback binary vector.

(2)

For a given circuit at discrete time τ:

$$y_j(\tau)=y_j(x(\tau),\ z(\tau-1)),\ j=1,\dots,k,$$
$$z_r(\tau)=z_r(x(\tau),\ z(\tau-1)),\ r=1,\dots,s,$$

where y_j and z_r are Boolean functions of (m+s) variables, which are specified by the structure of the circuit. A test $T^{(m,N)} = (t^{(1)},\dots,t^{(N)})$ is a sequence of N binary m - dimensional vectors $t^{(g)} = (t_1^{(g)},\dots,t_m^{(g)})$, g=1,...,N, which are applied to the input lines of the circuit. In the absence of input faults, obviously, $x(g)=t^{(g)}$.

We denote by $Y^{(k,N)}=(y^{(1)},\dots,y^{(N)})$ the sequence of output vectors produced by application to the circuit a sequence of test patterns $T^{(m,N)} = (t^{(1)},\dots,t^{(N)})$, provided that the memory was cleared before testing. For a fault-free device $Y^{(k,N)}$ is uniquely determined by $T^{(m,N)}$. However, if a fault f occurs in the circuit, $Y^{(k,N)}$ may depend on the fault: $Y^{(k,N)}=Y^{(k,N)}(f)$. Denote by f_0 the situation when circuit is fault-free.

Definition 2. Test $T=T^{(m,N)}$ detects a fault f in a given circuit, if for this circuit

$$Y^{(k,N)}(f) \neq Y^{(k,N)}(f_0). \tag{3}$$

Consider now a set of faults $F=\{f_w\}$, $w=0,1,\ldots, W$, $W=|F|-1$, which may occur in the circuit.

Definition 3. Test $T=T^{(m,N)}$ detects all the faults from set F in the circuit if for any $f_w \in F$, $w\neq 0$,

$$Y^{(k,N)}(f_w) \neq Y^{(k,N)}(f_0) \quad . \tag{4}$$

Definition 4. Test $T=T^{(m,N)}$ locates all the faults from set F in the circuit c if for any f_v, $f_w \in F, v\neq w$,

$$Y^{(k,N)}(f_v) \neq Y^{(k,N)}(f_w). \tag{5}$$

We will present now formal definitions for universal tests detecting or locating the given set of faults.

Let us consider a set of circuits $C=\{c_n\}$, $n=1,\ldots,M$, and a probability space (E,B,P), where E is the space of elemntary events $E=\{e_n\}$, the elementary event e_n is the application of a given test T to the circuit c_n, B is a Borel field on E, (in ur case B is the set of all subsets of E) and P is a probability measure on E, defined by the probabilities of elementary events $p(e_n)$. For a given set of faults F there exists an element $B_{det}(F)$ (respectively $B_{loc}(F)$) of the Borel field B which is the set of all the elementary events e_n such that (4) (respectively, (5)) is satisfied for the circuit c_n. Then the corresponding probabilities

$$P\{B_{det}(F)\} = \sum p(e_n), \quad e_n \in B_{det}(F) , \tag{6}$$

$$P\{B_{loc}(F)\} = \sum p(e_n), \quad e_n \in B_{loc}(F), \tag{7}$$

have a meaning of probabilities that the test T detects (respectively, locates) all the faults from set F being applied to a circuit chosen randomly from C according to probability distribution $p(e_n)$. Henceforth we shall assume that $C = C_{seq}$ is the set of all possible (m,k,s) - circuits, i.e., each circuit c_n C_{seq} is a realization of (k+s) (not necessarily distinct) Boolean functions of (m+s) input variables, and exactly one circuit corresponds to any possible ordered (k+s)-tuple of Boolean functions. Thus,

$$|C_{seq}| = M = 2^{(k+s)2^{m+s}} \quad .$$

We assume also that the probability distribution is uniform: $p(e_n) = {}^1/_M$. Then $P\{B_{det}(F)\}=P_{det}(T,F,m,k,s,)$ and $P\{B_{loc}(F)\} = P_{loc}(T,F,m,k,s)$ are equal to the fractions of all the (m,k,s) - circuits in which test T detects (respectivley, locates) all the faults from set F.

Now let m, k and s be variables which take on positive integer values. Consider a sequence of sets $(C_{seq}=C(m,k,s))$, where m takes on increasing integer values, and a sequence of corresponding tests $(T^{(m,N)})$, where N is a function of m.

Definition 5. A sequence of tests $(T^{(m,N)})$ is called universal for detection (respectively, for location) of all faults from set $F = F(m,k,s)$, if

$$\lim_{m\to\infty} P_{det}(T,F,m,k,s)=1 \tag{8}$$

(respectively, if

$$\lim_{m\to\infty} P_{loc}(T,F,m,k,s)=1\ ). \tag{9}$$

Obviously, (8) and (9) imply that the limit exists.

(For the sake of simplicity, we shall say also 'universal tests' instead of 'universal sequence of tests'). It is seen from Definition 5 that the performance of universal tests becomes better, the larger the VLSI devices under test. This asymptotic property of universal tests makes this approach especially relevant for complex VLSI circuits.

The following notations will be used throughout the chapter:

b - multiplicity of faults.

(T) - test matrix formed by test patterns from a test set T as rows.

N(F,m,k,s) - minimum number of test patterns in universal tests for detection of all faults from F.

$\varepsilon(a)$ - an arbitrary function such that $\varepsilon(a)\to\infty$ when $a\to\infty$.

For stuck-at faults we will use a construction of universal sequence of test patterns in which test $T = (t^{(1)},\dots, t^{(N)})$ of size N has the following structure:

$$t_i^{(1)} = 0, \quad i = 1,\dots,m;$$
$$t_i^{(3)} = 0, \quad i = 1,\dots,\lceil m/2\rceil,$$
$$t_i^{(3)} = 1 \quad i = \lceil m/2\rceil + 1,\dots,m\ ,$$

$$t_h^{(2h+3)}=1,\ t_i^{(2h+1)}=0,\ i\neq h,\ i=1,\dots,m,\ h=1,\dots,\frac{N}{2} - 2\ ; \tag{10}$$

$$t^{(2h)} = \overline{t^{(2h-1)}},\ h=1,\dots,\frac{N}{2} \quad (N\leq 2m+4)\ .$$

Here $\bar{t}$ is a binary vector which is the complement (negation) of t. Note that the order of test patterns (indicated by the upper index) is essential in the case of networks with memory.

Example 1. Let m=10, N=8. Then

$$(T) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix} .$$

C. Classes of Faults to Be Considered.

We shall start with input/output (terminal) faults for two reasons. First, in many practical cases, interconnections between chips in a computer system are less reliable than the chips themselves. During certain phases of production (e.g., soldering or wire wrapping connections) of digital circuits and systems, and also in integrated circuits themselves (especially in those circuits that were earlier found to meet specifications), faults are more likely to occur at the terminals of integrated circuits and circuit cards rather than inside the circuits [56]. (See also [39, 40, 57].) Moreover, a good understanding has been achieved and final results have been obtained for some classes of these faults.

In view of this, we shall consider here 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. The set of all such faults of multiplicity at most b is denoted by $F_b^{(is)}$

2. Input bridgings, when at most b input lines may be bridged. Bridging (short-circuit) faults have become increasingly important for VLSI devices [39-44]. 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. Since for any given type of technology only one type of bridging (either AND or OR) may appear in the device, we shall consider only AND-type bridgings between any two input lines. Of course, all results may be easily reformulated also for the case of OR-type bridgings. The set of all such faults of multiplicity at most b is denoted by $F_b^{(ib)}$

3. 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. The set of all such faults is denoted by $F^{(os)}$.

4. Output bridgings. In this case we consider all bridgings between any number of output lines. The set of all such faults is denoted by $F^{(ob)}$

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 [39 - 40]. However, we will not use these phenomena for detection of feedback bridgings, since their observation may present technical difficulties. The set of all feedback bridging faults is denoted by $F^{(fb)}$.

Universal testing of internal faults is a complicated problem because of its generality and because the concepts of controllability and observability come into play in a more involved manner than for terminal faults. (Terminal faults can be considered as a very special case, when we have either complete controllability (input faults), or complete observability (output faults).)

We shall illustrate how universal tests can be applied to detection of internal faults by two following examples .

1) Stuck-at faults at internal lines of combinational circuits

Denote by $F_{b,r}^{(ins)}$ the set of all faults of multiplicity at most b which can occur at r internal lines of a combinational circuit. The universal testing of these faults will be considered under specific assumptions concerning the ensemble of devices under test.

2) Stuck-at faults in memory cells of sequential circuits

Denote by $F_b^{(mem)}$ the set of all stuck-at faults of multiplicity at most b in memory cell of an (m,k,s) - circuit. This class of faults deserves special analysis, since in many practical situations memory cells are less reliable than gates.

D. Universal Testing of Terminal Faults in Combinational Circuits.

In this section we shall consider how the concept of universal testing has been applied to the problem of detection of input/output ('terminal') faults in combinational circuits.

Denote by $\alpha = \alpha(T,F)$ a lower bound on the fraction of test patterns in a test T, distorted by any fault from a given class F of input faults.

The following general theorem holds for any input faults.

Theorem 1. Let F be a set of any input stuck-at and/or bridging faults in combinational (m,k,o)-circuits. Then for any test T,

$$P_{det}(T,F,m,k,0) \geq 1 - W \cdot 2^{-\alpha Nk} \quad , \qquad (12)$$

$$P_{loc}(T,F,m,k,0) \geq (1 - \binom{W+1}{2} \cdot 2^{-\alpha Nk}), \text{ where } W=|F|-1. \qquad (13)$$

Proof. For the ensemble C(m,k,0) of all combinational (m,k,0)-circuits the probability that the output vectors are different from each other for two different faults (including f_0, i.e. the fault-free case) if the input vectors for a given test pattern become diferent as a result of these two faults is 2^{-k}. Thus, the probablitiy to distinguish between two given faults (including f_0) by application of a test in which at least α N test patterns are distorted by any fault (except f_0) is

$$q \leq 2^{-\alpha Nk} \quad . \qquad (14)$$

Hence, by use of the union bound, we obtain inequalities (12) and (13) for the probabilities of detection and location of all the faults from F, respectively.

1. Detection of Input Stuck-at Faults.

Theorem 2. The minimum number $N(F_b^{(is)},m,k,0)$ of test patterns in a universal sequence of tests which detects all the faults from $F_b^{(is)}$ in (m,k,0)-circuits satisfies an asymptotical inequality:

$$N(F_b^{(is)},m,k,0) \lesssim 2 \left\lceil \frac{\log_2 \sum\limits_{i=1}^{b} \binom{m}{i}}{k} \right\rceil + 4 \quad . \tag{15}$$

(As usual, $a(m) \lesssim c(m)$, if $\lim\limits_{m\to\infty} \frac{a(m)}{c(m)} \leq 1$; if $a(m) \lesssim c(m)$ and $c(m) \gtrsim a(m)$, then $a(m) \sim c(m)$, $\lceil a \rceil$ is the smallest integer greater than or equal to a.)

Proof. Consider a sequence of tests, in which the test $T = T^{(is)}(m,N)$ of length N is given by (10).

$$\text{Let } N = 2 \left\lceil \frac{1}{k} (\log_2 \sum_{j=1}^{b} \binom{m}{j} + \varepsilon(m) \right\rceil + 3 \quad . \tag{16}$$

For any unidirectional fault with multiplicity at most (i.e. for such a fault that the faulty inputs are either all stuck-at-0 or all stuck-at-1) the probability $q^{(is)}(T,m,k,0)$ of not detecting this fault by test $T = T^{(is)}(m,N)$ defined by (10) is bounded by

$$q^{(is)}(T,m,k,0) \leq 2^{-Nk/2} \quad . \tag{17}$$

For any not unidirectional fault

$$q^{(is)}(T,m,k,0) \leq 2^{-(N-3)k} \quad . \tag{18}$$

The probability P_{det} $(T, F_b^{(is)}, m, k, 0)$ that all the faults from $F_b^{(is)}$ are detected by the test $T = T^{(is)}(m,N)$ can be estimated by use of the union bound (the union bound is given by the fact that the probability of a union of events does not exceed the sum of probabilities of those events):

$$P_{det}(T, F_b^{(is)}, m, k, 0) \geq$$

$$1-2^{-Nk/2+1} \sum_{j=1}^{b} \binom{m}{j} - 2^{-(N-3)k} \sum_{j=1}^{b} (2^j-2) \binom{m}{j} \quad . \tag{19}$$

It follows from (16) and (19) that

$$\lim_{m \to \infty} P_{det} \; (T, F_b^{(is)}, m, k, 0) = 1 \tag{20}$$

for any $\varepsilon(m) \to \infty$ when $m \to \infty$. Thus, inequality (15) is proved.

Note that $P_{det}(T, F_b^{(is)}, m, k, 0)$ converges to 1 very fast with the increase of $|T|=N$.

Example 2. For single stuck-at faults ($b=1$) and one output ($k=1$) we obtain by (19): P_{det} $(T, F_1^{(is)}, m, 1, 0) \geq 1-2^{1-N/2}m$ (see also Fig. 2). For $m=16$, $N=30$ $P_{det}(T, F_1^{(is)}, 16, 1, 0)$ $\geq 1-2^{-10} \approx 99.9\%$ Thus, a test of the following form detects single input stuck-at faults in at least 99.9% of all combinational devices with 16 inputs and one output.

$$(T) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1 \end{bmatrix} \qquad \text{(30 rows)} \quad .$$

Corollary 1. For the set $F_m^{(is)}$ of input stuck-at faults of any multiplicity

$$N(F_m^{(is)},m,k,0) \lesssim 2(\left\lceil \frac{m}{k} \right\rceil + 2) \quad . \tag{21}$$

Example 3. For b=m, k=m, taking N=6 (by formula (21)), we obtain: $P_{det}(T,F_m^{(is)},m,m,0) \geq 1 - 3^{m}\cdot 3^{-2m}$, which for m=k=16 gives $P_{det} \geq 1-2.5\cdot 10^{-9}$. The test

$$(T) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

detects all input stuck-at faults of any multiplicity in almost all the devices with 16 input and 16 output lines, except very tiny fraction of $2.5\cdot 10^{-9}$.

Corollary 2. For $b < m/2$

$$N\ (F_b{}^{(is)},m,k,0) \lesssim 2 \left\lceil \frac{m}{k} \right\rceil H(\frac{b}{m} \quad , \qquad (22)$$

where $H(a) = -a\log_2 a - (1-a)\log_2(1-a)$.
Formula (22) follows immediately from (15) in a view of inequality

$$\log_2 \sum_{j=1}^{b} \binom{m}{j} \leq mH(\tfrac{b}{m})$$

for $b \leq m/2$.

Thus, the minimum number of test patterns increases at most linearly with the number of input lines and decrease inversely proportionally with the number of output lines (observation points). If $\frac{k}{m} \gtrsim H(\frac{b}{m})$, then for a large m two test patterns $t^{(1)} = (0,\dots,0)$ and $t^{(2)} = (1,\dots,1)$ detect all stuck-at faults of multiplicities at most b in almost all devices.

The use of the concepts of almost all devices needs some justification. Indeed, it implies that universal testing methods divide a given set of circuits into two subsets: circuits in which all the faults of a given class are detectable by the test (let us call them 'sheep'), and those which are not completely testable ('goats'). It might happen that, though the latter ones constitute only an infinitesimally small fraction of the total set, practically we are dealing just with these 'exceptional', 'atypical' circuits. In fact, a similar situation is not at all unusual in coding theory, logic design, etc. (One may call it 'Shannon paradox', since it appears, in particular, in information theory). To illustrate this point, let us consider a relevant example from the theory of complexity of Boolean functions.

It is well known that for almost all Boolean functions of m arguments, the minimal number G(m) of two-input gates in networks implementing these functions is $G \sim r \cdot 2^m \cdot m^{-1}$ (r is a constant) [58], but not even one class of Boolean functions which require for their implementation more than a polynomial number of gates is known yet.

The second example relates to Shannon's basic coding theorem which states that almost all block codes of a given length provide data transmission rate arbitrarily close to the channel capacity with probability of error tending to zero when the length increases. But in spite of the fact that all but an infinitesimally small fraction of codes are 'good', not a single class of such codes was known until quite recently [59].

Fortunately, this is not the case in universal testing. A number of examples considered below will demonstrate applicability of universal testing to some practically important classes of circuits. Though examples of 'goats' can be indicated among practical ones, they seem to be relatively rare (see Table I).

Example 4. Consider detection of input stuck-at faults with any multiplicity in an n-bit combinational adder. In this case m=2n, k=n+1. Formula (21) gives in this case N=8 for $m \longrightarrow \infty$. In fact, even for a finite m, only two test patterns are sufficient: universal test $T^{(m,2)} = (0, 1)$ detects all input stuck-at faults. It is easy to see that the same is valid for decoders, and any code converters (N=6 by (21)). For substractors (N=8 by (21)) and multipliers (N=6 by (21)) four test patterns are, in fact, sufficient.

Comparison with random testing illustrates the advantage of universal tests. Indeed, to detect only all single input stuck-at faults in the above-mentioned circuits, each column of the test matrix (T) must include at least one 0 and at least one 1. (Rows of (T) are test vectors of a random test T). For a random test of size N (assuming that all possible input vectors have equal probabilities to be chosen as test patterns) the probability of this event is

$$P_R(m,N) = (1-2\cdot 2^{-N})^m, \tag{23}$$

and $\lim_{m\to\infty} P_R(m,N) = 1$ only if $N \geq \log_2 m$.

(instead of N=2 for detection of all multiple input stuck-at faults by universal tests). The significant difference in numbers of test patterns demonstrated by this simple example has an obvious reason: the universal tests were developed specifically for this class of faults (stuck-at faults at input lines). There are good grounds to believe that the situation will be similar for other classes of faults, too (see, e.g., section II.D.2 in this chapter).

Example 5. Consider an up-and-down counter with parallel load as a combinational device with k=m-2 outputs, the first two inputs being control lines. Suppose that the counter adds 1 to the input number if the control signal is 10, substracts 1 if the control word is 01, and leaves the input unchanged if the control lines are on 00 or 11. Then the test given by (10) with N=8 test patterns detects all input stuck-at faults of any multiplicity. For instance, for m=8 the test is

$$(T) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

Indeed, all the stuck-at faults at data lines only are detected by the first and the second test patterns, combinations of single stuck-at-one faults at control lines with any faults at data lines - by the first, the second and the third test pattern, single stuck-at-zero at control lines and any faults at data lines - by the first, the second and the fourth test pattern,and double stuck-at faults at control lines combined with any faults at data lines are detected by the last four test patterns. The same universal test is also good for shifters.

The characterization of some standard combinational circuits regarding their testability by universal tests for the class of all input stuck-at faults of any multiplicity is given in Table I. (A circuit is called a 'sheep'(s) if universal tests of size N specified above provide 100% coverage of faults of a given class, and it is called a 'goat' (g) otherwise).

Table I shows that 'sheep' are more typical among standard hardware components than 'goats'. Of course, it should be born in mind that a 'sheep' with respect to one class of faults may be a 'goat' with respect to another class.

Another important feature of universal tests is that they can be easily extended so that many goats (as devices #17,21,23 in Table I) become 'sheep' with respect to a larger universal test. Of all the devices in the table only the multiplexer (#16)

TABLE I. Testability of standard computer components by universal tests.

s - 'sheep', g - 'goat'.

#	Name of the circuit	Function[1] $y(x_1,\ldots,x_m)$	Number of outputs k	Number of test patterns N	Testability
1.	AND gate	$x_1 \wedge x_2 \wedge \ldots \wedge x_m$	1	2m+4	s
2.	OR gate	$x_1 \vee x_2 \vee \ldots \vee x_m$	1	2m+4	s
3.	NAND gate	$\overline{x_1 \wedge x_2 \wedge \ldots \wedge x_m}$	1	2m+4	s
4.	NOR gate	$\overline{x_1 \vee x_2 \vee \ldots \vee x_m}$	1	2m+4	s
5.	Parity checker	$x_1 \oplus x_2 \oplus \ldots \oplus x_m$	1	2m+4	s
6.	Code convertors (BCD to binary, binary to Grey code, etc.)	$f(x_1,x_2,\ldots x_m)$ where f is one-to-one function	k=cm c-constant	$2\lceil 1/c\rceil+4$	s
7.	Comparator[2]	1 if $x^{(1)} \geq x^{(2)}$ 0 if $x^{(1)} < x^{(2)}$	1	2m+4	s
8.	Match detector[2]	1 if $x^{(1)} = x^{(2)}$	1	2m+4	s
9.	Adder[2]	$x^{(1)} + x^{(2)}$	k=m/2+1	8	s
10.	Subtractor[2]	$x^{(1)} - x^{(2)}$	k=m/2+1	8	s

11.	Multiplier[2]	$x^{(1)} \cdot x^{(2)}$	$k=2m$	6	s
12.	Shifter	$(x_3,\ldots,x_m)$ if $x_1=x_2$ $(x_4,\ldots,x_m,0)$ if $x_1=1, x_2=0$ $(0,x_3,\ldots,x_{m-1})$ if $x_1=0$, $x_2=1$	$k=m-2$	8	s
13.	Counter with parallel load	$(x_2,\ldots,x_m)$ if $x_1=0$ $(x_2,\ldots,x_m)+1 \pmod{2^{m-1}}$ if $x_1=1$	$k=m-1$	8	s
14.	Up-and-down counter with parallel load	$(x_3,\ldots,x_m)$ if $x_1=x_2$ $(x_3,\ldots,x_m)+1 \pmod{2^{m-2}}$ if $x_1=1$, $x_2=0$ $(x_3,\ldots,x_m)-1 \pmod{2^{m-2}}$ if $x_1=0$, $x_2=1$	$k=m-1$	8	s
15.	Decoder	$y_i(x_1,\ldots x_m)=1$ iff $(x_1,\ldots x_m)=i$ $(i=0,1,\ldots,2^m-1)$	$k=2^m$	6	s
16.	Multiplexer	$y(c,x) = x_{i+1}$ iff $c=i$ $(i=0,1,\ldots 2^r-1,\ r+2^r=m)$ $c=(c_1,\ldots,c_r)$, $x=(x_1,\ldots,x_{2^r})$	1	2m+4	g

(continued)

TABLE I (continued).

#	Name of the circuit	Function[1] $y(x_1,\ldots,x_m)$	Number of outputs k	Number of test patterns N	Testability
17.	Priority encoder	$y=i$ iff $2^i \leq x < 2^{i+1}$	$k=\lceil \log_2 m \rceil$	$2\left\lceil \frac{m}{\lceil \log_2 m \rceil} \right\rceil + 4$	g^3
18.	Hamming weight calculator	$\sum_{i=1}^{m} x_i$	$k=\lceil \log_2 m \rceil$	$2\left\lceil \frac{m}{\lceil \log_2 m \rceil} \right\rceil + 4$	s
19.	Hamming distance calculator	$\sum_{i=1}^{m/2} (x_i \oplus x_{i+m/2})$	$k=\lceil \log_2 m \rceil - 1$	$2\left\lceil \frac{m}{\lceil \log_2 m \rceil - 1} \right\rceil + 4$	s
20.	Decoder for an error-detecting code (cf. [59])	$y=s_1 \vee s_2 \vee \ldots \vee s_r$ where $(s_1,\ldots s_r) = (x_1,\ldots x_m)H$ and H is a check matrix	1	2m+4	s
21.	Decoder for an error-correcting code (cf. [59])	$y_i=x_i \oplus v_i$, where $(v_1,\ldots,v_m)$ is the code word closest to $(x_1 \ldots x_m)$ in Hamming metric	k=m	6	g^3
22.	Decoder for an arithmetic error-detecting AN-code (cf. [60])	$y(x)=0$ iff $x=cA$ $(c=0,1,2,\ldots)$, where A is the code generator.	1	2m+4	s

23.	Decoder for an arithmetic error-correcting AN code (cf [60])	$y=x-v$, where v is the code number closest to x in arithmetic metric	$k=m+1$	6	g[3]

[1] Here we do not distinguish between a binary vector and its numerical value in binary number system.

[2] For this device, we denote $x^{(1)}=(x_1,\dots,x_{m/2})$, $x^{(2)}=(x_{m/2+1},\dots,x_m)$

[3] This device becomes a 'sheep' if the universal test is extended to its maximum size of 2m+4 test patterns.

remains a 'goat', if the universal test of the maximum size (2m+4 test patterns) is applied..

Since practically we deal with finite values of m, the probability of complete fault coverage given by (19) is of a special interest for us. One can see that is converges to 1 very fast: the fraction of 'goats' decreases exponentially with the number test patterns N for a given m. For input stuck-at faults

$$1-P_{det}(T,F_b^{(is)},m,k,0) \simeq 2^{-Nk/2\ +1}\sum_{i=1}^{b}\binom{m}{i}2^i\ . \qquad (24)$$

Thus, adding ten more test patterns will decrease the fraction of 'goats' circuits by a factor of 32, if k=1, and by a factor of 1024, if k=2. Some values of this probability for k=1 are plotted in Fig.2.

2. Input Bridging Faults

Let us consider now the problem of detection of bridging faults between input lines.

By an input bridging of multiplicity b we mean any bridging (either AND or OR type) between at most b input lines. Denote the set of all such bridgings by $F_b^{(ib)}$. Denote by n(m,d) the length of a shortest error correcting code V with at least m distinct codewords and distance d. (We remind that a binary code V (n,d) of length n and distance d is a set of binary n-dimensional vectors which differ one from another in at least d components. Methods for design of such codes, as well as upper and lower bounds on n(m,d) are given, for instance in [59, 60].

Theorem 3. The minimum number $N(F_b^{(ib)}, m,k,0)$ of test patterns in a universal sequence of tests which detects all the

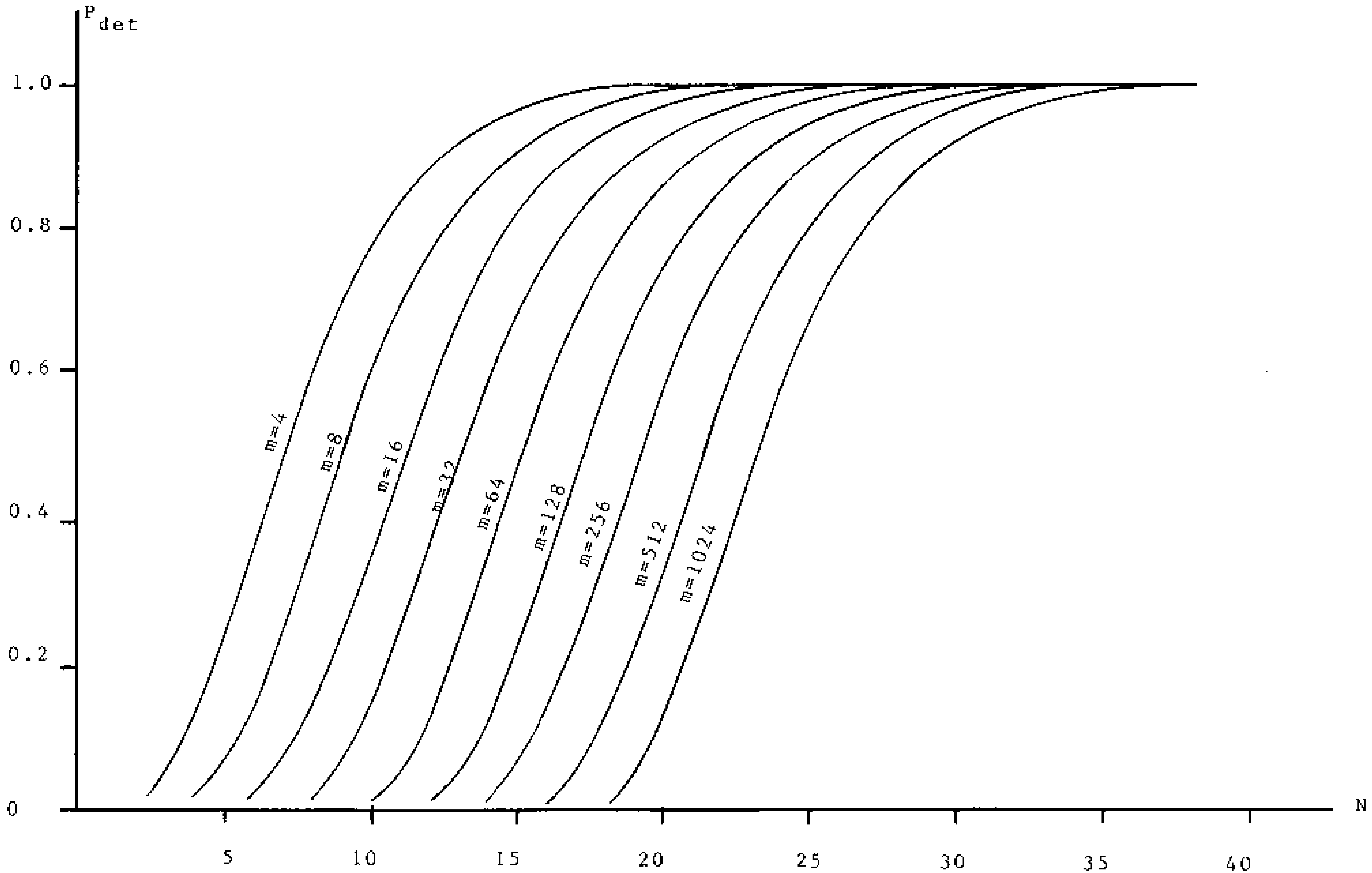

Fig. 2. The probability of detection of single input stuck-at faults in combinational (m,1,0)-devices as a function of the number of test patterns.

faults from $F_b^{(ib)}$ in (m,k,0)-circuits satisfies an asymptotic inequality:

$$N(F_b{}^{(ib)},m,k,0) \lesssim n(m, \left\lceil \frac{1}{k} \log_2 \sum_{j=2}^{b} j^{-k} \binom{m}{j} \right\rceil) \qquad . \qquad (25)$$

Proof. Denote by $(T_V(m,N))$ a binary matrix whose rows are test patterns and columns are distinct codewords of a binary error-correcting code V which has at least m codewords and distance d. Then there exists test $T_V(m,N)$ with the number N of test patterns equal to n(m,d). Choose the distance

$$d = \left\lceil \frac{1}{k} (\log_2 \sum_{j=2}^{b} j^{-k} \binom{m}{j} + \varepsilon(m)) \right\rceil \qquad , \qquad (26)$$

$$\text{where} \quad \varepsilon(m) \to \infty \text{ and } \varepsilon(m)(\log_2 \sum_{j=2}^{b} j^{-k}\binom{m}{j})^{-1} \to 0 \text{ as } m \to \infty \quad . \qquad (27)$$

Any bridging with multiplicity exactly j distorts at least n(j,d) test patterns. As well known (e.g., [59]), by the Singleton bound

$$n(j,d) \geq \log_2 j+d-1 \qquad (28)$$

Taking into account (27) and using the union bound, we obtain for the probability $P_{det}(T_V,F_b{}^{(ib)},m,k,0)$:

$$P_{det}(T_V,F_b{}^{(ib)},m,k,0) \geq 1-\sum_{j=2}^{b} \binom{m}{j} 2^{-n(j,d)k} \geq$$

$$1-2^{-(d-1)k} \sum_{j=2}^{b} j^{-k} \binom{m}{j} \quad . \qquad (29)$$

It is seen that, with d defined by (26),

$$\lim_{m\to\infty} P_{det}(T_V,F_b{}^{(ib)},m,k,0) = 1 \ . \tag{30}$$

Then formula (25) follows from (26) and (28) since

$$n\left(m,\left\lceil \frac{1}{k}\left(\log_2\sum_{j=2}^{b} j^{-k}\binom{m}{j}\right) + \varepsilon(m)\right\rceil\right) \sim n\left(m,\left\lceil \frac{1}{k}\log_2\sum_{j=2}^{b} j^{-k}\binom{m}{j}\right\rceil\right). \tag{31}$$

Example 6. Let V be the (8,4) extended single error-correcting Hamming code [59]. For this code d=4, n=8 and m=16. Consider this code as a test T_V (16,8) for a circuit with 16 inputs and k=4 outputs.

$$(T_V(16,8)) = \begin{bmatrix} 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&1&1&1&1&0&0&1&1&0&0&0&0&1&1 \\ 0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1 \\ 0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1 \\ 0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1 \end{bmatrix}$$

It follows from (29) that this test detects all bridgings between two input lines (b=2) in at least 99.8% of all combinational devices with 16 input and 4 output lines.

Corollary 3. If $$k > \log_2 \sum_{j=2}^{b} j^{-k} \binom{m}{j}, \tag{32}$$

then $$N(F_b^{(ib)},m,k,0) \sim \lceil \log_2 m \rceil \quad . \tag{33}$$

Proof. For any b $N(F_b^{(ib)},m,k,0) \geq \lceil \log_2 m \rceil$, since columns of T should be distinct. On the other hand, if (30) is valid, we can take d=1 and then $n(m,1) \sim \lceil \log_2 m \rceil$.

We note that in accordance with Corollary 3, universal tests with $N = \lceil \log_2 m \rceil$ and d=1 can be used for detection of all bridging faults (of any multiplicity) in adders, subtractors, multipliers, decoders, counters wih parallel load, and other devices for which the condition (30) is fulfilled.

Let us compare this result with the number of test patterns in a random test. Obviously, all the columns of (T) must be distinct in order to detect all single bridgings. The probability that a test chosen completely randomly satisfies this condition is

$$P_R(m,N) = \prod_{r=1}^{m-1} (1 - r \cdot 2^{-N})$$

and $\lim_{m \to \infty} P_R(m,N) = 1$ only if $N \geq 2 \log_2 m$. Thus, random testing requires at least twice more test patterns than universal testing in this case.

Corollary 4. For k=1, b=2 (single bridging faults between two lines)

$$N(F_2^{(ib)}, m,1,0) \leq 7.08 \log_2 m \quad . \tag{34}$$

Proof. By (25)

$$N(F_2^{(ib)},m,1,0) \leq n(m,\lceil \log_2 \frac{m(m-1)}{4} \rceil) \quad .$$

According to Varshamov-Gilbert bound [59] the function $n = n(m,d)$ satifies the inequality

$$\frac{\log_2 m}{n} \geq 1-H(\frac{d}{n}) \quad . \tag{35}$$

In our case

$$d=\lceil \log_2 \frac{m(m-1)}{4} \rceil \sim 2\log_2 m \quad .$$

Denote $(2/n_0) \log_2 m=x$, where n_0 turns (35) into equality. Obviously, $n_0 \geq n$. The equation

$$\frac{x}{2} = 1-H(x)$$

has a root $x \simeq 0.2825$. Thus,

$$n \leq \frac{2}{x}\log_2 m \simeq 7.08 \log_2 m,$$

which proves the corollary.

Example 7. Suppose m=16, k=1. Let the test matrix T_V be the Hadamard matrix [59] with all zeros row omitted.

$$(T_v) = \begin{bmatrix}
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\
0&0&1&1&0&0&1&1&1&1&0&0&1&1&0&0\\
0&1&0&1&0&1&0&1&1&0&1&0&1&0&1&0\\
0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0\\
0&1&0&1&1&0&1&0&0&1&0&1&1&0&1&0\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0\\
0&0&1&1&1&1&0&0&1&1&0&0&0&0&1&1\\
0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1\\
0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1\\
0&1&1&0&1&0&0&1&0&1&1&0&1&0&0&1\\
0&1&1&0&1&0&0&1&1&0&0&1&0&1&1&0
\end{bmatrix}$$

In this case $N = 15$, $d = 8$ and formula (29) yields $P_{det}(T_v,F_2^{(ib)},m,k,0) \geq 1 - 15.2^{-13} = 0.998$, which means that our test detects input bridgings with $b=2$ in 99.8% of all combinational devices with 16 inputs and 1 output line.

3. Output Stuck-at and Bridging Faults

The case of output faults seems to be easier to treat since in this case the lines which may be faulty coincide with observation lines. This fact leads to the result that universal tests for output faults degenerate to random tests [36]. The following theorem holds for $C=C(m,k,0)$ - the ensemble of all $(m,k,0)$-circuits.

Theorem 4. (i) The probability $P_{det}(T,F^{(os)},m,k,0)$ of the detection of all output stuck-at faults by any test T of size N is

$$P_{det}(T,F^{(os)},m,k,0) = (1-2^{-(N-1)})^k \tag{36}$$

(for any m).

(ii) The minimum number of test patterns for detection of all output stuck-at faults in almost all (m,k)-circuits, when $k \to \infty$, is asymptotically equal:

$$N(F^{(os)},m,k,0) \sim \log_2 k \quad . \tag{37}$$

(iii) The probability $P(T,F^{(ob)},m,k,o)$ of the detection of all output bridging faults by any test T of size N is

$$1-(k-1)2^{-N} \geq P_{det}(T,F^{(ob)},m,k,0) \geq 1-(\tfrac{k}{2})2^{-N} \tag{38}$$

(independently on m).

(iv) The minimum number of test patterns for detection of all output bridging faults in almost all (m,k)-circuits satisfies asymptotically ($k \to \infty$) the following inequalities:

$$\log_2 k \lesssim N(F^{(ob)},m,k,0) \lesssim 2\log_2 k. \tag{39}$$

Proof. Both stuck-at-zero and stuck-at-one faults at a given output line will be detected by a test of size N with the exception of two case only if the output for N test patterns are all zeros or are all ones. Thus, the probability of detecting stuck-at faults at a given line for the ensemble of circuits C(m,k,o) is $p = 1 - 2 \cdot 2^{-N}$, which leads to formula (36) for the probability of the detection fo all stuck-at faults at k output lines. Then (37) follows immediately from (36) in the asymptotic case ($k \to \infty$).

To prove (38) note that the probability that the bridging between two given output lines will not be detected by a test of size N is $q = 2^{-N}$. Then, by use of the union bound, we obtain the right hand part of (38). On the other hand, the probability to detect only the bridgings of a given line with all the others is $p = 1-(k-1)2^{-N}$, which yields the left-hand part of (38). Then (39) follows immediately for $k \longrightarrow \infty$.

Example 8. Let k=32 and N=16. Then for any m and any test T with $|T|=N=16$

$$P_{det}(T,F^{(os)},m,32,0) > 1-2^{-10} = 0.990,$$
$$P_{det}(T,F^{(ob)},m,32,0) > 1-31\ 2^{-12} = 0.9924.$$

4. Feedback bridging faults

Consider now single bridgings of AND type (OR-bridgings can be analyzed similarly) between an input and output line. A universal sequence of tests for this class of faults $F^{(fb)}$ consists of tests with the following test patterns:

$$t_i^{(1)} = 0, \quad i = 1,\dots,m,$$
$$t_{g-1}^{(g)}=1, \; t_i^{(g)}=0, \; i \neq g-1, \; i=1,\dots,m, \; g=2,\dots,N, \; (N\leq m) \quad . \quad (40)$$

(The test patterns are, in fact, the odd-numbered test vectors of the test defined by (10) except the third test vector in (10)).

Theorem 5. (i). The probability $P_{det}(T,F^{(fb)},m,k,0)$ of the detection of all single feedback bridgings by the test defined by formula (38) of size N satisfies the following bounds:

$$(1-2^{-N})k \geq P_{det}(T,F^{(fb)},m,k,0) \geq (1-2^{-(N-1)})k \qquad (41)$$

(for any m).

(ii). If $m \geq \log_2 k$ and $k \to \infty$, then the minimum number of test patterns for detection of all single feedback bridgings in almost all (m,k,o) -circuits is

$$N(F^{(fb)},m,k,0) \sim \log_2 k \qquad . \qquad (42)$$

Proof. For any test of size N the probability p to detect an AND bridging between any input and a given output line satisfies the inequality $p \leq 1-2^{-N}$. On the other hand, for the test defined by (40), $p \geq 1-2^{-(N-1)}$. This yeilds (41) and (42) (The condition $m \geq \log_2 k$ reflects the fact that $N \leq m$).

Example 9. For k=32 and $m \geq N=16$

$$P_{det}(T,F^{(fb)},m,32,0) > 1-32\cdot 2^{-15} = 0.9990,$$

and the test

$$(T) = \left[\begin{array}{cccccccccccccccccccccccc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0
\end{array}\right] \quad \begin{array}{l}(16\\ \text{rows})\end{array}$$

detects all single feedback bridgings in 99.9% of all combinational devices with 24 input and 32 output lines.

5. Detection of all single input/output faults

In this section we shall give the estimation of a minimum number $N(F^{(I/O)},m,k,0)$ of test patterns for the detection of all single input/output stuck-at, bridging and feedback bridging faults, or, more exactly, all faults from

$$F^{(I/O)} = F_1^{(is)} \cup F_2^{(ib)} \cup F^{(os)} \cup F^{(ob)} \cup F^{(fb)}$$

(Here $A \cup B$ means the union of sets A and B).

As in the previous section, we suppose that $k \rightarrow \infty$ and $m \geq \log_2 k$. Note that this condition is satisfied for most standard computer components, for example, for shifters, counters with the parallel load, adders, subtractors, multipliers, dividers, etc., we have $k = rm$ where $0.5 \leq r \leq 1$. It is also satisfied for many standard PLAs. (The important exceptions from this rule are multiplexers and decoders/demultiplexers.)

Comparing the results from Sections II.D.1 - II.D.4, we can observe that the detection of bridging faults is a more difficult problem than the detection of stuck-at faults. With this in mind, we shall, in this section, try to use the tests developed for the detection of bridging faults for the detection of stuck-at faults. (This approach is opposite to that used in [42, 43] where tests developed for detection of stuck-at faults were also used for detection of bridging faults.)

Let L be a linear code with distance $\lceil \frac{2}{k}(\log_2 m+\varepsilon(m)) \rceil$ and length at least $\lceil \log_2 k \rceil$ containing at least m+2 different codewords. Then L contains at most one vector v with a number of zeros less than $\lceil \frac{1}{k} (\log_2 m+\varepsilon(m))|$. Let $T_L=T_L(m,k)$ be the test such that columns of (T_L) are codewords of L distinct from all zeros and v. Then, if $m \geq \log_2 k$ and $k \longrightarrow \infty$, the tests $T_L(m,k)$ form a universal sequence of tests for detection of all faults from $F^{(I/O)}$.

Theorem 6. The minimum number of test patterns for detection of all single input/output stuck-at and bridging faults in almost all (m,k)-circuits when $m \geq \log_2 k$ and $k \longrightarrow \infty$ satisfies asymptotically the following inequalities:

$$\max[\log_2 m, \log_2 k] \leq N(F^{(I/O)}, m, k, 0) \leq \log_2 k + \max[\log_2 m, \log_2 k]\,. \tag{43}$$

Proof. The inequalities follow from (39), (42) and from the fact that $N(F_2^{(ib)}, m, k, 0) \geq \lceil \log_2 m \rceil$ (to detect all input bridgings, all columns of (T) must be distinct).

We note also that for such important (from a practical point of view) devices as shifters, counters with the parallel load, adders, subtractors, multipliers and dividers we have $k=rm$ $(0.5 \leq r \leq 1)$, and minimum numbers of test patterns for the detection of input/output stuck-at and bridging faults are in fact, between $\log_2 k$ and $2 \log_2 k$ [39, 40], which illustrates the practical usefulness of estimations given by formula (43).

Since the general expression for the probability of complete fault detection by universal tests T_L is rather cumbersome, we shall present here only some numerical results (Table II).

It follows from Table II that, to provide the probability of detecting all single terminal faults at least 99.9%, it is sufficient to apply between 25 and 55 test patterns for any device with $30 \leq m \leq 510$ input lines and $1 \leq k \leq 512$ output lines.

TABLE II.[1] Minimum of test patterns for detection of all single input/output stuck-at and bridging faults with a probability $P_{det}(T_L, F^{(I/O)}, m, k, 0) > 1-2^{-10} = 0.9990$.

m k	30	62	126	254	510
1	54	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

E. Universal Testing of Stuck-at Faults at Internal Lines of Combinational Circuits .

Consider now the problem of detection of stuck-at faults at internal lines of combinational circuits.

Denote input variables of a combinational circuit by $x_1,\ldots,x_m$, output variables by $y_1,\ldots,y_k$ and variables at r internal lines by $u_1,\ldots,u_r$. Suppose that $u_i = \varphi_i(x_1,\ldots,x_m)$, $(i=1,\ldots,r)$, and $y_j = \psi_j(u_1,\ldots,u_r)$, $(j=1,\ldots,k)$, $\varphi_i \in B_m$, $\psi_j \in B_r$, where B_m and B_r are the sets of all Boolean functions of m and r arguments respectively. Consider an ensemble of devices such that all possible sets of functions $\{\varphi_i\}$, $(i=1,\ldots,r)$ and $\{\psi_j\}$, $(j=1,\ldots,k)$ appear with the same probability $P = |B_m|^{-r} \cdot |B_r|^{-k}$.

[1] For computations in Table II the tables of the best codes from [59], Appendix A1, have been used.

Denote by $F_{b,r}^{(ins)}$ the set of all stuck-at faults of multiplicity at most b which may occur at the r internal lines under consideration.

Theorem 7. The minimum number $N(F_{b,r}^{(ins)},m,k,0)$ of test patterns in a universal sequence of tests which detects all the faultsfrom $F_{b,r}^{(ins)}$ satisfies an asymptotic inequality (as $r \to \infty$ and b is fixed):

$$N(F_{b,r}^{(ins)},m,k,0) \lesssim -\left\lceil \frac{b \log_2 r}{\log_2(2^{-k}+2^{-b}-2^{-(k+b)})} \right\rceil \quad . \tag{44}$$

Proof. Consider a random test T with

$$N = -\left\lceil \frac{b \log_2 r + \varepsilon(r)}{\log_2(2^{-k}+2^{-b}-2^{-(k+b)})} \right\rceil \tag{45}$$

test patterns where $\varepsilon(r) \to \infty$ as $r \to \infty$. The probability of detecting any stuck-at fault at internal lines of multiplicity exactly j by any one test pattern is

$$P_j(T,m,k,0) = (1-2^{-k})(1-2^{-j}) = 1-2^{-j}-2^{-k}+2^{-(j+k)} \quad .$$

Using the union bound, we obtain the following inequality for the probability $P_{det}(T,F_{b,r}^{(ins)},m,k,0)$ of detecting all stuck-at faults of multiplicity at most b at internal lines by a random test T of length N:

$$P_{det}(T,F_{b,r}^{(ins)},m,k,0) \geq 1 - \sum_{j=1}^{b} 2^j \binom{r}{j} (2^{-j}+2^{-k}-2^{-(j+k)})^N \ . \tag{46}$$

It is seen from (45) and (46) that

$\lim_{r\to\infty} P_{det}(T,F_{b,r}^{(ins)},m,k,0) \rightarrow 1$ which proves the theorem.

One can see from (45) that the probability P_{det} converges to 1 very fast with the increase of the number r of internal lines (exponentially in r if $s(r) \sim \log_2 r$)

F. Applications of Universal Tests to Sequential Circuits.

In this section we shall consider the problem of detection of stuck-at faults at imput lines and at memory cells in sequential circuits with a bloc-diagram represented by Fig.1. (It should be pointed out, to avoid misunderstanding, that circuits with the memory depth larger than 1 are represented by the block-diagram at Fig. 1 as well: they correspond to special cases when some of the functions at the s feedback lines do not depend on the m primary inputs).

1. Detection of input stuck-at faults.

Universal testing of terminal faults in sequential circuits is a natural generalization of the approach applied to the terminal faults in combinatorial circuits. Yet it poses a number of quite non-trivial new problems, since the temporal behavior of sequential circuits is much more complex. In particular, the time order of test patterns becomes essential in this case.

Note that introduction of memory in a device may lead only to a decrease of a number of test patterns required for detection of input stuck-at faults. Indeed, some input faults which distort input test patterns but do not distort the corresponding output vectors at time t may distort information in memory, and this may result in distortions of primary output vectors at times t+1, t+2, etc. (This conclusion may seem to be counterintuitive, since functional testing of sequential circuits is more difficult and requires longer tests than that for combinational circuits. Nevertheless, this is not the case for universal testing).

Detection of single input stuck at faults in sequential (m,k,s)- circuits has been considered in [38]. The generalization of this result for the case of input stuck-at faults of any multiplicity is presented below. In contrast with combinational devices, the order of test patterns is essential for devices with memory. The same test (10) that has been used for testing input stuck-at faults in combinational devices can be used for sequential devices as well.

Theorem 8. The minimum number $N(F_b^{(is)},m,k,s)$ of test patterns in a universal sequence of tests that detects all the faults from $F_b^{(is)}$ in sequential (m,k,s)-devices satisfies an asymptotical ($m \to \infty$) inequality:

$$N(F_b^{(is)},m,k,s) \leq \max \Bigg\{ 2 \left\lceil \frac{\log_2 \sum_{j=1}^{b} \binom{m}{j}}{k - \log_2(2^{-k}+2^{-s}-2^{-(k+s)})} \right\rceil +2 \;,$$

$$\frac{1}{k} \log_2 \sum_{j=1}^{b} (2^j-2)\binom{m}{j} +4 \Bigg\} \quad . \qquad (47)$$

Proof. Take T = T(m,N) defined by (10)

$$\text{with } N = \max\left\{2\left\lceil \frac{\log_2 \sum_{j=1}^{b} \binom{m}{j} + \varepsilon(m)}{k-\log_2(2^{-k}+2^{-s}-2^{-(k+s)})} \right\rceil ,\right.$$

$$\left.\left\lceil \frac{1}{k}\left(\log_2 \sum_{j=1}^{b}(2^j-2)\binom{m}{j} + \varepsilon(m)\right)\right\rceil + 3 \right\}, \tag{48}$$

where $\varepsilon(m) \to \infty$ as $m \to \infty$.

Consider first detection of unidirectional stuck-at faults. It is easy to see that for a test (10) a test pattern which is not distorted by a fault of this type, always follows a test pattern which is distorted by the same fault. If the test pattern is distorted by a fault, then the fault will not be detected at the same moment of time (when the test pattern is applied) with probability 2^{-k}. If the test pattern is not distorted by a fault, then the fault will not be detected at the same moment of time if one of two events occur: 1) the memory input was not distorted by the fault at the previous moment of time, probability of this event is 2^{-s}, or 2) the memory input was distorted at the previous moment of time, but this did not result in distortion of the output vector at the present moment of time, probability of this event is $(1-2^{-s})2^{-k}$. Since any unidirectional fault distorts at least N/2 test patterns in test (10), the probability q_1 that such a fault will not be detected by the test is upperbounded by:

$$q_1 \leq 2^{-Nk/2}(2^{-k}+2^{-s}-2^{-(k+s)})^{N/2} \quad . \tag{49}$$

Any nonunidirectional fault distorts at least N-2 test patterns, and the probability q_2 that such a fault will not be detected by the test is upperbounded by:

$$q_2 \leq 2^{-(N-3)k} \quad . \tag{50}$$

Hence, the probability that all faults form $F_b^{(is)}$ are detected by the test can be estimated by use of the union bound as follows:

$$P_{det}(T, F_b{}^{(is)}, m, k, s) \geq 1 - 2^{-Nk/2}(2^{-k}+2^{-s}-2^{-(k+s)})^{N/2} \cdot 2\sum_{j=1}^{b} \binom{m}{j} - \\ - 2^{-(N-2)k} \sum_{j=1}^{b} (2^j-2)\binom{m}{j} \quad . \tag{51}$$

It is seen from (47) and (51) that

$$\lim_{m \to \infty} P_{det}(T, F_b{}^{(is)}, m, k, s) = 1, \tag{52}$$

which proves the theorem.

Theorem 2 follows from Theorem 8 if s=0. Moreover, the following bounds are valid.

Corollary 5.

$$N\ (F_b{}^{(is)}, m, k, s) \leq 2(\left\lceil \frac{1}{k} \log_2 \sum_{j=1}^{b} \binom{m}{j} \right\rceil + 2), \tag{53}$$

if $s/k \to 0$ as $m \to \infty$.

Corollary 6. For the set $F_m^{(is)}$ of all input stuck-at faults with any multiplicity,

$$N(F_m^{(is)}, m, k, s) \leq 2\left\lceil \frac{m}{2k} \log_2 3 + 3 \right\rceil + 4 \qquad (54)$$

if $s/k \rightarrow 1$, as $m \rightarrow \infty$

Example 10. Consider detection of stuck-at faults in an adder-accumulator with m input lines. In this case $k=s>m$, and the test $T^{(m,2)}$ with test patterns $t^{(1)}=(0,\ldots,0)$, $t^{(2)}=(1,\ldots,1)$ detects all the faults, which agrees with Theorem 8.

The probability $P_{det}(T, F_b^{(is)}, m, k, s)$ converges to 1 very fast with increase of m and N, as the following examples show.

Example 11. For $s=k=m/2$, $b=m$, $N=6$,
$P_{det}(T, F_m^{(is)}, m, m/2, m/2) \geq 1 - 3^m \cdot 2^{-2m}$.

Example 12. If $m = s = 16$, $k = 1$, $b = 1$, then $P_{det}(T, F_1^{(is)}, 16, 1, 16) \geq 1-2^{-5}$ for $N=10$ and $P_{det}(T, F_1^{(is)}, 16, 1, 16) \geq 1-2^{-5}$ for $N=20$. Some numerical values of $P_{det}(T, F_1^{(is)}, m, k, s)$ are plotted in Figure 3, for $k=s=1$.

2. Detection of stuck-at faults in memory

Consider now detection of stuck-at faults in memory cells of sequential (m,k,s)-circuits. Such a fault is said to be of multiplicity j $(1 \leq j \leq s)$ if j outputs of memory cells are stuck.

Theorem 9. The minimum number $N(F_b^{(mem)}, m, k, s)$ of test patterns in a universal sequence of tests which detects all the faults from the set $F_b^{(mem)}$ of memory stuck-at faults of multiplicity at most b satisfies an asymptotic inequality (as

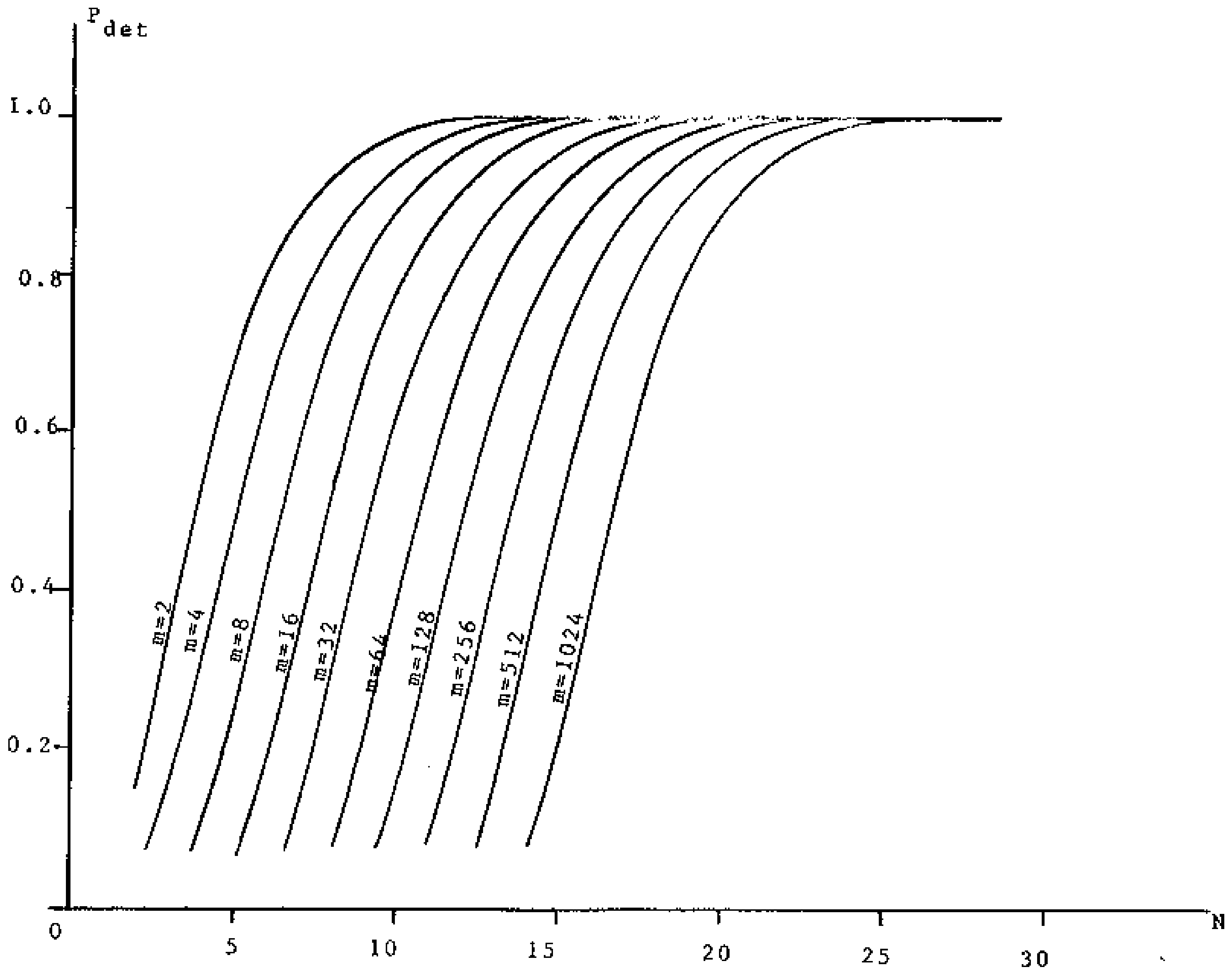

Fig. 3. The probability of detection of single input stuck-at faults in sequential (m,1,1)-circuits as a function of the number of test patterns.

$s \to \infty$, for fixed b and any m):

$$N(F_b^{(mem)},m,k,s) \leq \left\lceil \frac{b \log_2 s}{b - \log_2(1+2^{b-k}-2^{-k})} \right\rceil \quad . \qquad (55)$$

Proof. Consider a random test T with N test patterns. The probability q that a stuck-at fault in memory of multiplicity exactly j will not be detected by any one test pattern is the sum of probabilities of two events: 1) the memory output is not distorted by the fault, the probability of this event is 2^{-j}, or 2) the memory output is distorted by the fault, but the distortion is 'masked' by the combinational part of the circuit, the probability of this event is $(1-2^{-j})2^{-k}$. Hence

$$q = 2^{-j} + (1-2^{-j})2^{-k} \quad . \qquad (56)$$

Thus the lower bound for the probability $P_{det}(T,F_b^{(mem)},m,k,s)$ of detecting all faults from $F_b^{(mem)}$ by the test T with N test patterns is given by:

$$P_{det}(T,F_b^{(mem)},m,k,s) \geq 1 - \sum_{j=1}^{b} 2^j \binom{s}{j} (2^{-j} + 2^{-k} - 2^{-(j+k)})^N . \qquad (57)$$

$$\text{For} \quad N = \left\lceil \frac{b \log_2 s + \varepsilon(s)}{b - \log_2(1+2^{b-k}-2^{-k})} \right\rceil , \qquad (58)$$

where $\varepsilon(s) \to \infty$ as $s \to \infty$, it follows from (57) that

$$\lim_{s \to \infty} P_{det}(T,F_b^{(mem)},m,k,s) = 1 \quad , \qquad (59)$$

which proves the theorem.

For $k \gg b$ (55) gives:

$$N \lesssim \log_2 s \quad . \tag{60}$$

In particular, for detection of single stuck-at faults in memory

$$P_{det}\ (T, F_1^{(mem)}, m, k, s) \geq 1-2s(1+2^{-k})^N 2^{-N} \quad . \tag{61}$$

Example 13. For k=s=16, $P_{det}(T,F_1^{(mem)},m,16,16) \geq 1-2^{-15}$ for N=20, and $P_{det}(T,F_{16}^{(mem)},m,16,16) \geq 1-2^{-15}$ for N=40.

It can be shown that the asymptotic results (37) and (39) for detection of output stuck-at and bridging faults remain valid for sequential circuits. Taking into account also (49) and (60) we come to the following proposition:

Corollary 7. For detection of all stuck-at and bridging faults at input and output lines and all stuck-at faults in memory in sequential circuits with m = k = s.

$$N(F,m,m,m) \lesssim 2\log_2 m \quad . \tag{62}$$

It can be shown [38] that *location* of input stuck-at faults and memory faults in (m,k,s) - circuits requires universal tests of double size as compared with universal tests for *detection* of those faults.

G. Detection of a Fraction of Faults by Universal Tests

In previous sections we supposed that in almost all devices 100% of faults from a given class F were to be detected by universal tests. In this section we shall consider a weaker requirement that in almost all devices only at least $(1-\beta)W$ faults are to be detected, where $0<\beta<0.5$ and $W=|F|-1 \dashrightarrow \infty$. In this case we shall use the same universal tests as in the previous sections. As we shall see below, for any constant $\beta>0$ the minimal number of test patterns $N^{(\beta)}(F,m,k,s)$ is not increasing with an increase of m. Let $P^{(\beta)}_{det}$ (T,F,m,k,s) be a probability of detection of $\lceil(1-\beta)W\rceil$ faults from the set F by the given test T in a randomly chosen (m,k,s)-device.

Theorem 10. (i) Let F_b be a set of any input stuck-at or bridging faults with multiplicity at most b in combinational (m,k,0)-devices. Then for any test T

$$P^{(\beta)}_{det}(T,F_b,m,k,0) \geq \sum_{i=0}^{\lfloor \beta W \rfloor} \binom{W}{i} (1-2^{-\alpha Nk})^{W-i}\, 2^{-\alpha Nki}. \tag{63}$$

For input stuck-at faults in combinational networks,

$$N^{(\beta)}(F_b{}^{(is)},m,k,0) \leq 2(1 - \left[\frac{\log_2 \beta}{k}\right]). \tag{64}$$

and for input bridging in combinational networks

$$N^{(\beta)}(F_b{}^{(ib)},m,k,0) \leq n(m,(1- \left[\frac{\log_2 \beta}{k}\right]). \tag{65}$$

(n(m,d) has been defined in Section II.D.2. By $\lfloor a \rfloor$ we denote the largest integer such that $\lfloor a \rfloor \leq a$.)

(ii) Let $F = F_b^{(is)}$ be the set of input stuck-at faults with multiplicity at most b in sequential (m,k,s) - devices. Then, for test T defined by (10),

$$P_{det}^{(\beta)}(T,F_b^{(is)},m,k,s) \geq \sum_{i=0}^{\lfloor \beta W \rfloor} \binom{W}{i} (1-2^{-Nk/2}(2^{-s}+(1-2^{-s})2^{-k})^{N/2})^{W-i} \cdot$$

$$2^{-Nki/2}(2^{-s}+(1-2^{-s})2^{-k})^{Ni/2} \quad , \tag{66}$$

$$N^{(\beta)}(F_b^{(is)},m,k,s) \leq 2(1 + \left\lceil \frac{\log_2 \beta}{\log_2(2^{-s}+(1-2^{-s})2^{-k})-k} \right\rceil) \tag{67}$$

if $\quad 2^{-Nk/2}(2^{-k}+2^{-s}-2^{-(k+s)})^{N/2} \geq 2^{-(N-3)k}$,

and

$$P_{det}^{(\beta)}(T,F_b{}^{(is)},m,k,s) \geq \sum_{i=0}^{\lfloor \beta W \rfloor} \binom{W}{i} (1-2^{-(N-3)k})^{W-i} 2^{-(N-3)ki} \quad , \tag{68}$$

$$N^{(\beta)}(F_b{}^{(is)},m,k,s) \leq 4-2 \left\lfloor \frac{\log_2 \beta}{2k} \right\rfloor \tag{69}$$

otherwise.

Proof. Denote by $\lambda = \lambda(m,k,s)$ the probability of detection of any given fault f F by test T. Then, by definition,

$$P_{det}{}^{(\beta)}(T,F,m,k,s) = \sum_{i=0}^{\lfloor \beta W \rfloor} \binom{W}{i} \lambda^{W-i}(1-\lambda)^{i} \quad . \tag{70}$$

For input stuck-at or bridging faults in combinational devices we have, from Theorem 1,

$$\lambda = \lambda(m,k,0) \geq 1-2^{-\alpha Nk} \quad \text{for any } m. \tag{71}$$

For input stuck-at faults in sequential (m,k,s)-devices it follows from Theorem 8 that for a test defined by (10),

$$\lambda=\lambda(m,k,s)\geq \min[1-2^{-Nk/2}(2^{-s}+(1-2^{-s})2^{-k})^{N/2},1-2^{-(N-3)k}] \quad (72)$$

Formulas (63), (66) and (68) now immediately follow from (70), and (71), and (72). It follows from (70) that $P_{det}^{(\lambda)}(T,F,m,k,s)\rightarrow 1$ iff $\beta > 1-\lambda$ for $W \longrightarrow \infty$. Thus, in view of (71), if for a combinational device

$$\alpha N > -(1/k)\log_2\beta \quad , \qquad (73)$$

then $\beta > 1-\lambda$ and $P_{det}{}^{(\beta)}(T, F_b{}^{(is)},m,k,s) \longrightarrow 1$ as $W \longrightarrow \infty$. Formulas (64) and (65) now immediately follow from (73).
For sequential devices, if N is equal to the right-hand parts of inequalities (67), or, respectively (69) then for a test defined by (10), in view of (72), $\beta > 1-\lambda$ and $P_{det}{}^{(\beta)}(T,F_b{}^{(is)},m,k,s) \rightarrow 1$ as $W \rightarrow \infty$.

As we can see from Theorem 10, minimum numbers of test patterns detecting any given fraction $1-\beta(0<\beta<0.5)$ of faults in almost all combinational devices do not depend on multiplicity of faults b and for stuck-at faults do not depend on numbers of input lines m. (As we have seen in previous sections, for $\beta=0$ minimum numbers of test patterns for detection of input stuck-at faults depend on m and b).

Example 14. Let us estimate a minimum number of test patterns detecting 99% of input stuck-at faults of any multiplicity in almost all devices. From (64) with $\beta = 0.01$ we have, at most, 14 test patterns are sufficient for combinational networks (if $k\geq 7$, then two test patterns are sufficient). For sequential networks, at most 12 test patterns will be sufficient for detection of single input stuck-at faults (if $k-\log_2(2^{-s}+2^{-k}\cdot(1-2^{-2}))\geq 7$, then two test patterns are sufficient).

Example 15. Consider combinational circuits with $m=16$ and $k=1$. Suppose we are interested in detecting single and double input stuck-at faults with a fault coverage of $1-\beta=99\%$. Then, by (64), $N=16$. The total number of faults $W=4\binom{m}{2}+2m=512$, and $\lfloor\beta W\rfloor=5$. Since for test (10) $\lambda\geq 1-2^{-Nk/2}$, formula (70) yields

$$P_{det}^{(0.01)}(T,F_2^{(is)},\ 16,\ 1,\ 0)\geq 0.995,$$

which means that the test detects 99% of all single and double stuck-at faults in 99.5% of all such devices.

A summary of the results on universal testing is presented in Table III.

TABLE III. Summary of results on universal testing

Class of circuits	Class of Faults	Probability of detection	Minimum number of Test patterns
C(m,k,o)	$F_b^{(is)}$	$P_{det}(T,F_b^{(is)},m,k,0)\geq$ $\geq 1-2^{1-Nk/2}\sum_{j=1}^{b}\binom{m}{j}-$ $2^{-(N-3)k}\sum_{j=1}^{b}(2^j-2)\binom{m}{j}$	$N(F_b^{(is)},m,k,0)\lesssim$ $\lesssim 2\lceil (1/k)\log_2\sum_{i=1}^{b}\binom{m}{i}\rceil+4$
C(m,k,s)	$F^{\ (is)}$	$P_{det}(T,F_b^{(is)},m,k,s)\geq 1-$ $-2^{1-Nk/2}(2^{-k}+2^{-s}-s^{-(k+s)})^{N/2}.$ $\sum_{j=1}^{b}\binom{m}{j}-2^{-(N-2)k}\sum_{j=1}^{b}(2^j-2)\binom{m}{j}$	$N(F_b^{(is)},m,k,s)\lesssim$ $\lesssim \max\{\ 2\left\lceil\frac{\log_2\sum_{j=1}^{b}\binom{m}{j}}{k-\log_2(2^{-k}+2^{-s}-2^{-(k+s)})}\right\rceil+2\ ,$ $\lceil 1/k\ \log_2\sum_{j=1}^{b}(2^j-2)\binom{m}{j}\rceil+3\}.$
C(m,k,o)	$F^{\ (ib)}$	$P_{det}(T_c,F_b^{(ib)},m,k,0)\geq$ $1-2^{(1-d)k}\sum_{j=2}^{b}\ j^{-k}\binom{m}{j}$	$N(F_b^{(ib)},m,k,0)\leq$ $n(m,\lceil (1/k)\log_2\sum_{j=2}^{b}\ j^{-k}\binom{m}{j}\rceil)$
C(m,k,s)	$F^{(os)}$	$P_{det}(T,F^{(os)},m,k,s)=$ $(1-2^{-(N-1)})^k$	$N(F^{(os)},m,k,s)\sim\log_2 k$

C(m, k, s)	$F^{(ob)}$	$1-(k-1)2^{-N}\geq P_{det}(T,F^{(ob)},m,k,s)\geq$	$\log_2 k \lesssim N_{ob}(F^{(ob)},m,k,s) \lesssim 2\log_2 k$
C(m, k, o)	$F^{(fb)}$	$1-\binom{k}{2}2^{-N}(1-2^{-N})^{k}\geq P_{det}(T,F^{(fb)},m,k,0)\geq$ $(1-2^{-(N-1)})^{k}$	$N(F^{(fb)},m,k,0)\sim\log_2 k$
C(m, k, o)	$F_{b,r}^{(ins)}$	$P_{det}(T,F_{b,r}^{(ins)},m,k,0)\geq$ $\geq 1-\sum_{j=1}^{b}2^{j}\binom{r}{j}(2^{-j}+2^{-k}-2^{-(j+k)})^{N}$	$N(F_{b,r}^{(ins)},m,k,0\lesssim$ $\lesssim -\left\lceil\frac{\log_2 r}{\log_2(2^{-k}+2^{-}-2^{-(k+b)})}\right\rceil$
C(m, k, s)	$F_{b}^{(mem)}$	$P_{det}(T,F_{b}^{(mem)},m,k,s)\geq$ $\geq 1-\sum_{j=1}^{b}2^{j}\binom{s}{j}(2^{-j}+2^{-k}-2^{-(j+k)})^{N}$	$N(F_{b}^{(mem)},m,k,s)\leq$ $\lesssim \left\lceil\frac{\log_2 s}{b-\log_2(1+2^{b-k}-2^{-k})}\right\rceil$

III. ADVANTAGES AND LIMITATIONS OF UNIVERSAL TESTING METHODS.

A. General Remarks.

As with any other method of testing, universal tests have their specific advantages and limitations. In this section we discuss briefly some general properties of universal tests, and then we compare this approach with other known methods.

B. Advantages.

1). Simple test generation. Since universal tests are developed for a wide class of circuits, it eliminaiates the necessity of generating tests individually for any particular device - the problem that becomes practically unsolvable for complex circuits, if we want to test them on the gate level.

2). Guaranteed full coverage for almost all devices. The estimation of fault coverage poses a difficult problem in functional testing. Universal tests are constructed in such a way as to provide a 100% coverage of faults for almost all devices from a broad set. It should be borne in mind, however, that this coverage is provided for a specific class of faults, for which this particular universal test was designed.

3). Applicability to complex circuits. The crucial question in universal testing is, of course, how large is the fraction of 'atypical' circuits which are not completely testable by a given universal test. Perhaps the most attractive feature of universal tests is that they are asymptotically efficient, i.e., they work the better the more complex is the circuit under test (e.g., the larger is the number of input and/or output faults). Therefore universal testing seems to have a good potential for VLSI testing.

4). Extendability. Sometimes we may be interested to extend the universal test in order to further diminish the risk to encounter a 'goat'. The universal tests designed above can be easily extended (up to the limit $N = 2m+4$) without any change in construction. As it was shown above, with the increase of the number of test patterns, the probability that the device under test is a 'goat' decreases exponentially. If one needs a number of test patterns $N>2m+4$ (which may happen in case when the multiplicity of faults b is almost equal to m, and k=1) test vectors of Hamming weight 2 and their negations can be added to the test.

5). Good coverage for 'goats'. One should not think that 'goats' are tested poorly by universal tests. The situation is just the opposite: in fact, although the coverage of faults in 'goats' is not complete, almost all 'goats' are covered almost completely. For example, if the total fraction of 'goats' is 1%, then in only 0.5% of them (i.e., in a $5 \cdot 10^{-5}$ fraction of all the circuits) two or more faults remain undetected, and in only a $1.7 \cdot 10^{-7}$ fraction of all the devices more than two faults are undetectable by the test. In general, it is shown in Section I.G. that, in contrast with the detection of all stuck-at faults, the detection of any fraction $1-\beta$ of the total number of faults (even if this fraction is arbitrarily close to 1, but remains constant) requires asymptotically a number of test patterns which depends on β only, but does not depend on m and b.

6). Another advantage of the approach presented - this time, one of the methodological nature - can be mentioned. Since, from the viewpoint of universal testing, classes of circuits are described probabilistically, this opens a way for applications of powerful mathematical techniques based on probability theory and theory of random processes (for sequential circuits) to testing problems. In particular, close relation between testing theory and information theory (including coding) can be already seen. (For example, universal tests for input bridging faults are based on optimal error - correcting codes.)

C. Limitations.

1). The major limitation of universal testing is that it becomes efficient only in the case when a broad spectrum of circuits is to be tested. This argument is applicable both for manufacturer and user testing. Indeed, if many copies of only a few types of devices are to be tested, there is reason to invest in the development of specific tests for each of the types, since a specific test may be shorter and may provide a sufficient coverage for just this particular type. Seemingly, the most justifiable application of universal tests is in the case of a user who has to test a large variety of devices which come in small numbers of copies. Moreover, there is a good reason to use universal tests as a first step in the testing procedure, since with probability close to 1 they will detect faulty devices, thereby eliminating the necessity of further testing.

2). Although universal tests are good for almost all devices, there is always a danger that a particular device under test is a 'goat'. Since universal testing is based on probabilistic reasoning, it provides good results only in the statistical sense and cannot guarantee against individual failures.

3). The size of universal tests may appear to be substantially larger than that of specific tests. Naturally, that is the price for their 'universality'. However, for many typical circuits (as we have seen it in the example of terminal faults in adders, subtractors, multipliers, decoders) the universal tests are not longer than specific ones.

Universal testing, by its nature, combines some feature of functional and random testing, filling the gap between them. Therefore, we shall compare in more details these two approaches with universal testing.

D. Universal vs. Functional Testing.

Functional testing requires complete knowledge of the functional description of the device under test, while universal testing based on general information only about the class of devices (such as number of input and output variables, some general features of circuit topology, etc.). Functional testing, being device-oriented, can provide shorter tests than the universal one, but has greater test generation complexity. It is usually very hard to estimate the fault coverage for functional tests. The fault coverage for universal tests is known in terms of probability that, in a device chosen from a given class, all (or a given fraction) of faults are detectable.

Let us illustrate the situation for the case of multiple input stuck-at faults. The problem of functional testing for this class of faults has been considered in [56, 57]. It has been shown in [56] that, if the number of inputs $m>5$, a specific test can be constructed for any such circuit which detects all input stuck-at faults and consist of not more than $2m-4$ test patterns. There exists an example of a function which requires at least $2(m-r)$ tests patterns, where $r \geq \log_2(m-r)$. It follows from the results described in Section II.D. that a universal test which consists of $N = 2m(1 + \delta)$ test patterns will detect all the input stuck-at faults in almost all possible devices, except for a fraction of $2^{-m\delta}$. On the other hand, even for this simple class of faults, functional test generation is much more complex than that for universal tests.

In the case of bridgings the situation is similar. It has been shown [61-67] that the number of test patterns in a functional test that detects all input bridgings between two lines grows linearly with the number of inputs m. According to Corollary 4, the number of test patterns in a universal test for the same class of faults grows as $\log_2 m$ only.

E. Universal vs. Random Testing.

At first glance, universal testing looks very similar to random testing. Indeed, both of them ignore specific features of the device under test, and the performance of both types of tests can be characterized in terms of probability. However, this similarity is misleading, since there are essential differences between these two approaches which lead to different results with regard to the size of tests and their fault-detecting capability.

1). Universal tests are usually designed in a deterministic manner, in which not only the test patterns but also their order (in case of sequential circuits) may be essential. On the other hand, for random testing, test patterns and their order are chosen randomly [2, 17-25, 68].

2). In contrast with random testing, universal tests are designed on the basis of information about a <u>class</u> of circuits. The class can be characterized by the numbers of input and output lines, feedback lines, memory cells, by the number of gates and/or internal lines, by characterization of the class implementef functions, e.g., self-dual or symmetric functions, by features of circuit topology, such as path complexity, number of gate levels (e.g., PLAs), etc.

3). Universal tests are fault-oriented: they are designed for a specific class of faults and depend on this class, as we have seen in the example of single and multiple input stuck-at faults and input bridging faults at input lines. It was shown, for instance, that one needs a universal test with two patterns only to detect all input stuck-at faults in such devices as adders, decoders, and code convertors, while a random test would require more than $\log_2 m$ test patterns. For the class of all input multiple bridging faults in adders, substractors, multipliers, decoders, shifter, counters with parallel load, etc., we have obtained $N \sim \log_2 m$ for universal tests (based on error-correcting codes with the distance depending on the multiplicity of faults), while a random test would require $N \geq 2 \log_2 m$ to detect all single input bridging faults. Thus, as a rule, universal tests are substantially shorter than random ones. In case of output faults, however, universal tests degenerate to random tests. As a matter of fact, the extent of 'universality' is rather flexible and depends on the 'width' of the class of circuits and the class of faults involved. If the classes are not specified, we come back to random testing, while a complete description of a device on functional or on gate level will lead to functional or gate-level testing, respectively. Thus, both device-specific and random testing may be considered as special limit cases of universal testing. The comparison of various types of testing is summarized in Table IV.

Finally, we may conclude that universal testing suggests a viable alternative to other types of testing and expands the arsenal of tools for a testing engineer.

TABLE IV. Comparative characterization of various approaches to testing.

Type of testing	Input data for test generation	Test generation complexity	Test size necessary for a given fault coverage	Fault coverage for a given test size
1. Gate level	Gate-level description of the device under test, class of fault	Maximal (Optimal test generation is NP-hard)	Minimal	Maximal
2. Functional	Functional description of the device under test, class of faults	Large	Larger than for a gate-level test	Lower than for a gate-level (often, difficult to estimate)
3. Universal	Description of a class of devices: class of faults	Small (the same test for the whole class of devices for a given set of faults)	Larger than for the first two types. (Depends on the class of devices and the set of faults)	High for almost all devices from a given class
4. Random	Set of possible test patterns only	Minimal	Maximal	Minimal

IV. CONCLUDING REMARKS.

Since the concept of universal testing is rather a new one, a lot of work is still ahead in this area. The main goal of the future research is to explore the applications of this approach to various classes of devices and fault models and to develop both general mathematical techniques and new universal test sequences for those classes.

The investigation of faults at internal lines should include a deeper analysis of the properties of controllability and observability, treated from the statistical point of view.

For example, circuits built with components which implement self-dual Boolean functions have complete controllability, since for a self-dual function $\bar{f}(\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_m) = f(x_1,x_2,\ldots,x_m)$. (Note that any Boolean function can be made self-dual by addition of one binary argument). This property can facilitate application of universal testing methods to the corresponding class of circuits.

Another important class of circuits to be investigated is the PLA. Various classes of PLAs can be defined in terms of probabilities of interconnections between the primary inputs and AND gates, and between the AND and OR gates. These parameters are expected to determine the characteristics of universal sequences of tests. More generally, various classes of circuits with given topology can be considered. (A trivial example is provided by fanout-free circuits, where testing of internal faults can be reduced to testing of terminal faults.)

Another important type of faults is that of faults at interconnections between microprocessors and/or other VLSI components forming a computer system. Since each of the components of the network is a complex circuit by itself, the universal testing approach seems to be relevant and promising for this type of faults. However, the investigation in this direction is complicated by a peculiar effect of 'degeneration' of Boolean functions: a class of networks built with components implementing uniformly distributed random Boolean functions, generally speaking, does not produce uniformly distributed Boolean functions by itself. A deeper probabilistic analysis seems to be necessary for application of universal testing methods to this set of faults.

REFERENCES.

1. Broussard, M.E. (1973). 'Higher Yields, Lower Costs', Proc. Int. Symp. on Fault-Tolerant Computing, June 1973, pp. 6-11.
2. M.A. Breuer, A.D. Friedman. (1976). 'Diagnosis and Reliable Design of Digital Systems', Computer Science Press.
3. Roth, J.P. (1966). 'Diagnosis of Automata Failure: A Calculus and a Method', IBM Journal of Research and Development, v. 10, July 1966, pp. 278-291.
4. Sellers, F.F., Hsiao, M.Y., and Bearnson, C.L. (1968). 'Analyzing Errors with the Boolean Difference', IEEE Trans. on Computers, v. C-17, July 1968, pp. 676-683.
5. Goel, P. (1980). 'Test Generation Cost Analysis and Projections', 17th Design Automation Conference Proceedings, Minneapolis, Minnesota, June 1980, p. 77-84.
6. Ibarra, A.K. and Sahni, S. (1975). 'Polynomially Complete Fault Detection Problems', IEEE Trans. on Computers, C-24, March 1975, pp. 242-248.
7. Sahni, S. and Bhatt, A. (1980). 'The Complexity of Design Automation Problems', Proc. 17th Design Automation Conference, Minneapolis, MN, May 1980, p. 402-411.

8. Saluja, K.K., Shen, Li, and Su, S.Y.H. (1983). 'A Simplified Algorithm for Testing Microprocessors', Proc. 1983 Test Conference, Philadelphia, 1983.

9. Thatte, S.M., and Abraham, J.A. (1980). 'Test Generation for Microprocessors', IEEE Trans. on Computers, v. C-29, n. 6, June 1980, pp. 423-441.

10. Abraham, J.A. and Parker, K.P. (1981). 'Practical Microprocessor Testing: Open Loop and Closed Loop Approaches', 11th Intern. Symposium on Fault-Tolerant Computing, 1981.

11. Akers, S.B. (1978). 'Functional Testing with Binary Decision Diagrams', The 8th Annual International Conference on Fault Tolerant Computing, Toulouse, France, June 1978.

12. Karpovsky, M.G. (1977). 'Error Detection in Digital Devices and Compter Programs with the Aid of Linear Recurrent Equations over Flinite Commutative Groups', IEEE Trans. on Computers, v. C-26, n. 3, pp. 208-219.

13. Karpovsky, M.G. and Trachtenberg, E.A. (1977). 'Linear Checking Equations and Error-Correcting Capabilty of Computation Channels', Proc. IFIP Congress, North Holland, pp. 619-624.

14. Goel, N., and Karpovsky, M.G. (1982). 'Functional Testing of Computer Hardware Based on Minimization of Magnitude of Undetected Errors', IEE Proc., v. 129, Sept. 1982, pp. 169-181.

15. Akers, S.B. (1978). 'Functional Testing with Binary Decision Diagrams', 8th Int. Symposium on Fault-Tolerant Computing, Toulouse, France, pp. 75-82.

16. Shedletsky, J.J. (1977). 'Random Testing: Practicality vs. Verified Effectiveness', The 7th Annual Intern. Conference on Fault-Tolerant Computing, Los Angeles, CA, June 1977, pp. 175-179.

17. Rault, J.C., (1971). 'A Graph Theoretical and Probabilistic Approach to the Fault Detection of Digital Circuits', International Symposium on Fault-Tolerant Computing, March 1971, pp. 16-29.

18. David, R. and Thevenod-Fosse, P. (1980). 'Minimal Detecting Transition Sequences: Application to Random Testing', IEEE Trans. on Computers, C-29, n. 6, June 1980, pp. 514-518.

19. Agrawal, V.D. and Agrawal, P. (1972). 'An Automatic Test Generation System for ILLIAC-IV Logic Boards', IEEE Trans. on Computers, v. C-21, pp. 1015-1017, Sept. 1972.

20. Agrawal, P. and Agrawal, V.D. (1975). 'Probabilistic Analysis of Random Test Generation Method for Irredudant Logic Networks', IEEE Trans. on Computers, v. C-24, pp. 681-685, June 1975.

21. Agrawal, P. and Agrawal, V.D. (1975). 'On Improving the Efficiency of Monte Carlo Test Generation', Int. Symposium on Fault-Tolerant Computing, Paris, France, pp. 205-208, June 1975.

22. Bostin, D., Girard, E., Rault, J.C. and Tulloue, R. (1973). 'Probabilistic Test Generation Methods', Int. Symposium on Fault-Tolerant Computing, Palo Alto, p. 171, June 1973.

23. Savir, Ditlow, and Bardell, P.H. (1983). 'Random Pattern Testability', International Symposium on Fault-Tolerant Computing, June 1983, Milano, Italy.

24. Thevenod, P. and David, R. (1983). 'Random Testing of Control Section of a Microprocessor', International Symposium on Fault-Tolerant computing, June 1983, Milano, Italy.

25. Thevenod-Fosse, P. and David, R. (1978). 'A Method to Analyze Random Testing of Sequential Circuits', Digital Processes, v. 4, pp. 313-332.

26. McCluskey, E.J., Bozorgui-Nesbat, S. (1981). 'Design for Autonomous Test', IEEE Trans. on Computers, v. C-30, pp. 866-875.

27. Savir, J. (1980). 'Syndrome-testable Design of Combinational Clircuits', IEEE Trans. on Computers, v. C-29, n. 6, June 1980, pp. 442-450.

28. Barzilai, Z., Coppersmith, D. and Rosenberg, A. (1981). 'Exhaustive Bit Pattern Generation in Discontiguous Positions with Applications to VLSI Self-testing', IBM Research Report, RC-8750, March 1981.

29. Chandra, A.K., Kou, L.T., Markowsky, G., Zaks, S. (1981). 'On Sets of Boolean n-Vectors with All k-Projections Surjective', IBM Research Report RC-8936, July 1981.

30. McCluskey, E.J. (1982). 'Built-in Verification Test', 1982 Intern. Test Conference, Nov. 15-18, pp. 183-190.

31. Tang, D.T., Chen, C.L. (1983). 'Logic Test Pattern Generation using Linear Codes', Proc. of 13th International Fault-Tolerant Computing Symposium, Milano, Italy, June 1983, pp. 222-226.

32. Tang, D.T., and Woo, L.S. (1982). 'Exhaustive Test Pattern Generation with Constant Weight Vectors', IBM Research Report, RC-9442, June 1982.

33. Tang, D.T., Chen, C.L. (1983). 'Efficient Exhaustive Pattern Generation for Logic Testing', IBM Research Report, RC-10064, July 1983.

34. Cohen, G., Karpovsky, M., Levitin, L.B. (1984). 'Exhaustive Testing of Combinational Devices With Outputs Depending on a Limited Number of Inputs', 1984 IEEE Information Theory Workshop, Caesarea, Israel, July 1-5, 1984.

35. Karpovsky, M.G. (1983). 'Universal Tests for Detection of Input/Output Stuck-at and Bridging Faults', IEEE Trans. on Computers, v. C-32, pp. 1194-1198.

36. Karpovsky, M.G. (1982). 'Universal Tests Detecting Input/Output Faults in Almost All Devices', Proc. of 1982 Int. Test Conference, Cherry Hill, NJ, Nov. 1982.

37. Karpovsky, M. and Levitin, L.B. (1983). 'Error Detection in Combinational Networks by Use of Codes Based on Hadamard Matrices', 1983 IEEE Inter. Symp. on Information Theory, St. Jovite, Quebec, Canada, Sept. 26-30, 1983.

38. Karpovsky, M.G., and Levitin, L.B. (1983). 'Detection and Identification of Input/Output Stuck-at and Bridging Faults in Combinational and Sequential VLSI Networks by Universal Tests', in Integration, The VLSI Journal, v. 1, 1983, p. 211-232.

39. Karpovsky, M. and Su, S. (1980). 'Detection and Location of Input and Feedback Bridging Faults Among Input and Output Lines', IEEE Trans. on Computers, C-29, June 1980, pp. 523-527.

40. Karpovsky, M. and Su, S. (1980). 'Detection of Bridging and Stuck-at Faults at Input Output Pins of Standard Computer Components', Proc. 17th Design Automation Conference, Minneapolis, MN, May 1980, pp. 494-506.

41. Mei, K.C.Y. (1974). 'Bridging and Stuck-at Faults', IEEE Trans. on Computers, v. C-23, pp. 720-727, July 1974.

42. Iosupovicz, A. (1978). 'Optimal Detection of Bridging Faults and Stuck-at Faults in Two-Level Logic', IEEE Trans. on Computers, C-27, May 19778, pp. 452-455.

43. Kodandapani, K.L. and Pradhan, D.K. (1980). 'Undectability of Bridging Faults and Validity of Stuck-at Fault Test Set', IEEE Trans. on Computers, v. C-23, pp. 55-59, January 1980.

44. Friedman, A.D. (1974). 'Diagnosis of Short-Circuit Faults in Combinational Circuits', IEEE Trans. on Computers, C-23, July 1974, pp. 746-752.

45. Eichelberger, E.B. and Williams, T.W. (1977). 'A Logic Design Structure for LSI Testability', Proc. 14th Design Automation Conference, June 1977, pp. 462-468.

46. Bennets, R.G. (1984). 'Design of Testable Logic Circuits', Addison-Wesley.

47. Shannon, C.E. and Weaver, W. (1949). 'The Mathematical Theory ofCommunication', Univ. of Illinois Press, Urbana, IL.

48. Levedev, D.S. , Levitin, L.B. (1966). 'Information Transmission by Electromagnetic Field, Information and Control', v.9, n. 1, 1-22.

49. Levitin, L. B. (1981). 'A Thermodynamic Characterization of Ideal Physical Information Channels', Journal of Information and Optimization Sciences, v. 2, n. 3, September 1981.

50. Levitin, L.B. (1982). 'Physical Limitations of Rate, Depth and Minimum Energy in Information Processing', International Journal of Theoretical Physics, v. 21, n. 2/3.

51. Shmukler, Yu. (1970). 'Unidimensional and Bidimensional Gur Games' Automation and Remote Control, n. 10, p. 1634.

52. Shmukler, Yu. (1971). 'Ballot Problem with a Random Error', DAN SSSR, v. 196, n. 4, p. 789.

53. Rosonoer, L.I. (1969). 'Random Logical Nets, I, II, and III', Automation and Remote Control, n. 5, pp. 773-781, n. 6, pp. 922-931, n. 7, pp. 1130-1139.

54. Fujiwara, H, Kinoshita, K. and Ozaki, H. (1980). 'Universal Test Sets for Programmable Logic Arrays', in Dig. 10th Inter. Symp. on Fault-tolerant Computing, Kyoto, Japan, June 1980, pp. 137-142.

55. Fujiwara, H., Kinoshita, K. (1981). 'A Design of Programmable Logic Arrays with Universal Tests', IEEE Trans. on Circuits and Systems, v. CAS-28, n. 11, Nov. 1981, pp. 1027-1032.

56. Kuhl, T.G. and Reddy, S.M. (1978). 'On Detection of Terminal Stuck-Faults', IEEE Trans. on Computers, C-27, May 1978, pp. 467-469.

57. Weiss, C.D. (1972). 'Bounds on the Length of Terminal Stuck-Fault Tests', IEEE Trans. on Compters, C-21, pp. 305-309.

58. Lupanov, O.B. (1955). 'On the Possibilities of Synthesis of Networks from Different Types of Elements', Dokl. Akad. Nauk. USSR, v. 103, pp. 561-563.

59. MacWilliams, F.J. and Sloane, N.J.A. (1977). 'The Theory of Error-Correcting Codes', North Holland.

60. Peterson, W.W. (1961). 'Error-Correcting Codes', M.I.T. Press, New York, London.

61. Sapoznikov, V.V. and Chikhonin, V.M. (1978). 'Detection of Short-Type Faults in Combinational Circuits', Automatika i Vychislitelnaya Technika, n. 6.

62. Dubrovskiy, A.V. (1980). 'Construction of a Unit Length Test for Testing a Combinational Circuit for Single Bridge-Type Faults', Automatika i Vychislitelnaya Technika, n.2.

63. Dubrovskiy, A.V. (1980). 'Construction of an Input Test Sequence to Check a Combinational Circuit for Single Bridge-Type Faults', Automatika i Vychislitelnaya Technika, n.5.

64. Asaf'ev, V.V. et al. (1981). 'Synthesis of Easily Controllable Circuits - A Direction Dictated by Integral Circuit Technology', Izv. AN SSSR, Tekhn. Kibernet., n. 4.

65. Goryashko, N.P. (1981). 'On the Synthesis of Circuits with Minimal Length of Test', Automation and Remote Control, n. 1.

66. Goryashko, N.P. (1982). 'Certain Results of the Theory of Synthesis of EasilyTestable Networks', Izv. An SSSR, Tekhn. Kibernet., n.2.

67. Goryashko, N.P. (1983). 'On the Synthesis of Combinational Circuits Which Are Easily Tested for Short Circuit-Type Faults', Engineering Cybernetics, v. 21, Sept.-Oct. 1983.

68. Huary, H., and Breuer, M.A. (1975). 'Analysis of Detectability of Faults by Random Patterns in a Special Class of NAND Networks', Computers and Elect. Engr., v.1, March 1975, pp. 77-84.