V. Conclusions

In this paper, we presented algorithms for the implementation of data transfer requirements of a system through indirect paths. This algorithm yeilds considerable reduction in bus interfaces over and above the minimal interface solution obtained through direct path realization [4]. Even though this technique was in use for a long time, no formal procedure for identifying the possible candidates for indirect path realization was reported. With the rapid advances in integrated circuit technology, the trend in digital design is to implement the complete system on a single chip. As module interconnections take a lion's share of the chip area, design efforts must be aimed at reducing these interconnections without sacrificing system speed specifications. Also, some researchers are recommending redundancy in interconnections [1] for improving the chip yield and reliability. In view of these developments, optimization of interconnections takes new dimensions in the system design discipline and the algorithm developed will serve as an effective design tool in cost effective realization of digital systems.

References

[1] T. E. Mangir, and A. Avizienis, "Fault-tolerant design for VLSI: Effect of interconnect requirements on yield improvement of VLSI Designs," *IEEE Trans. Comput.*, vol. C.31, no. 7, pp. 609-616, July 1982.

[2] A. Mathialagan and N. N. Biswas, "Optimal interconnections in the design of microprocessors and digital systems," *IEEE Trans. Comput.*, vol. C-29, no. 2, pp. 145-149, Feb. 1980.

[3] A. Mathialagan, "On the optimization of control memory and data paths in the design of micro-programmed computers and microprocessors," Ph.D. dissertation, Dep. Elect. Commun. Eng., Indian Institute of Science, Bangalore, Mar. 1980.

[4] S. S. S. P. Rao, "An integrated approach for the optimal design of digital systems towards VLSI realizations," Ph.D. dissertation Dep. Elect. Eng., Indian Institute of Technology, Bombay, India, 1983.

[5] H. C. Torng and N. C. Wilhelm, "The optimal interconnection of circuit modules in microprocessors and digital system design," *IEEE Trans. Comput.*, vol. C-26, no. 5, pp. 450-457, May 1977.

Design of Parity Testable Combinational Circuits

BHARGAB B. BHATTACHARYA and SHARAD C. SETH

Abstract—**The parity testability of a single output is related to its partition in terms of maximal supergates and then a scheme is proposed for making an untestable circuit parity testable by augmenting its maximal supergates. Only a small amount of extra logic and a single external *test-mode* pin is required to complete the design. The test procedure is simple and the hardware overhead is low.**

Index Terms—**Design for testability, combinational logic, parity testing, stuck-at-faults.**

I. Introduction

In this paper, we analyze the parity and the subparities of a combinational logic circuit. Our analysis is based on the unique minimum cover of the circuit defined by its maximal supergates [1], [2]. In these earlier papers, the supergate cover was used as a vehicle for computing probabilistic testability measures. Here, we show that it has wider applicability: the parities and the subparities of a circuit output are derivable from those of its maximal supergates. Our investigations lead us to suggest testable design schemes in which logic modifications are made at the (maximal) supergate level.

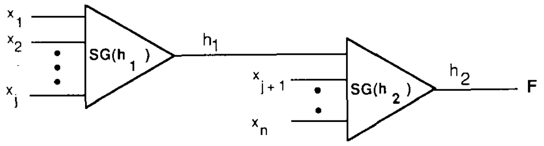

Fig. 1. Illustration of Theorem 1.

II. Parity Testing

Given an n-variable switching function $F(x_1, x_2, \cdots, x_n)$, the *(primary) parity* of F denoted as $p(F)$ is defined as follows:

$$p(F) = (\text{number of minterms of } F) \bmod 2.$$

It is well known that if the (primary) parity of the function is odd all multiple input faults can be detected by checking the primary parity alone [3]. Akers [4] extended the range of parity-testable circuits by examining additional parities associated with a function as follows.

Definition. The *parity-bit signature* (PBS) of an n-variable switching function $F(x_1, x_2, \cdots, x_n)$ is given by an $(n+1)$-bit binary vector:

$$\text{PBS}(F) = \langle p_0, p_1, p_2, \cdots, p_n \rangle$$

where

$$p_0 = p(F)$$
$$p_1 = p(\bar{x}_1 \cdot F)$$
$$p_2 = p(\bar{x}_2 \cdot F)$$
$$p_n = p(\bar{x}_n \cdot F).$$

In other words, p_i represents the parity of the subfunction $F(x_1, x_2, \cdots, x_i = 0, \cdots, x_n)$ that is, the parity of F when its ith variable is set to 0. PBS(F) consists of p_0, which we call the *primary parity* of F and n *secondary* or *subparities*, $p_1, p_2, \cdots, p_n$.

The parity-bit signature has been suggested for built-in self testing in [4].

A. Network Decomposition and Primary Parity

Let $F(x_1, x_2, \cdots, x_n)$ be the function realized by a single-output combinational network N whose maximal supergate partitions are SG(1), SG(2), $\cdots$, SG(k). Let F_i denote the function realized by SG(i) alone, i.e., when SG(i) is isolated from the rest of the circuit.

Theorem 1: $p(F)$ is odd if and only if the primary parities of all individual functions $F_1, F_2, \cdots, F_k$ are odd, i.e.,

$$p(F) = p(F_1) \cdot p(F_2) \cdots p(F_k).$$

Proof:[1] For simplicity, let us assume that circuit N has only two maximal supergates SG(h_1) and SG(h_2) as shown in Fig. 1. Primary inputs $x_1, x_2, \cdots, x_j$ feed SG(h_1), and the rest $x_{j+1}, \cdots, x_n$ feed SG(h_2). Let a and b denote the number of 0's and 1's appearing at h_1 when all 2^j combinations are exercised. Now let c and d denote

Manuscript received April 21, 1987; revised August 1, 1988 and April 10, 1989.

B. B. Bhattacharya is with the Electronics Unit, Indian Statistical Institute, Calcutta, India 700 035. This work was done while he was a visitor in the Department of Computer Science, University of Nebraska, Lincoln, NE 68588.

S. C. Seth is with the Department of Computer Science, University of Nebraska, Lincoln, NE 68588.

[1] Another proof of the theorem follows from Tokmen's disjunctive decomposition theorem [5].

the number of times 1 appears at primary output h_2, when h_1 is set to 0 and 1, respectively, and when all input combinations of $\{x_{j+1}, \cdots, x_n\}$ are exercised. Let F_1 and F_2 be the functions realized by the supergates SG(h_1) and SG(h_2), respectively.

Clearly, $a+b = 2^{j}\cdot (j > 0)$, hence both a and b must be either odd or even. Moreover, $p(F_1) = b \bmod 2$ and $p(F_2) = (c+d) \bmod 2$.

Since the set of primary input lines feeding SG(h_1) and SG(h_2) are disjoint, we have

$$p(F) = (ac + bd) \bmod 2.$$

[If part]: Assume $p(F_1) = 1 = p(F_2)$. Then both a and b are odd, and either c or d (but not both) must also be odd. Therefore, $p(F) = 1$.

[Only if part]: Assume $p(F_1) = 0$; then both a and b must be even implying $p(F) = 0$. On the other hand, if $p(F_2) = 0$, then c and d must be both even or both odd, which also implies that $p(F) = 0$.

To prove the general case, this argument can be applied recursively to any network having an arbitrary number of maximal supergates. ■

The following known result [3] is a corollary of Theorem 1.

Corollary 1. Any fan-out-free circuit of basic gates (that is, not including XOR and XNOR gates) must realize an odd number of minterms.

Proof: In a fan-out-free circuit, each basic gate AND, OR, NAND, NOR, NOT) is a maximal supergate by itself. The individual primary parity of a basic gate is always odd, and therefore, from Theorem 1, the global parity also becomes odd. ■

B. Network Decomposition and Subparities

The next theorem shows a relation of the subparities in a circuit with the parities of individual supergates.

Theorem 2: Let N denote a single output combinational network realizing $F(x_1, x_2, \cdots, x_n)$, with maximal supergate partitioning SG(1), SG(2), $\cdots$, SG(k). Let x_i be a primary input of N which is directly connected to supergate SG(j). Denote by F_h the function realized by the isolated supergate SG(h), for $h = 1, 2, \cdots, k$. Then the ith subparity of F is given by

$$p_i(F) = p_i(F_j) \cdot p(F_1) \cdot p(F_2) \cdots p(F_{j-1}) \cdot p(F_{j+1}) \cdots p(F_k).$$

Proof: The subparity $p_i(F)$ is obtained by observing the parity at the primary output when input x_i is set to 0. Since x_i is directly connected to supergate SG(j), it cannot be an input to any other supergate in a single output circuit. Therefore, setting $x_i = 0$ affects the functionality of supergate SG(j) alone. The rest of the proof follows immediately from Theorem 1. ■

III. Parity Testability of Stuck-at Faults

Given a circuit N, a fault in N is said to be *parity testable*, if the faulty parity-bit signature differs in at least one bit position from that of the fault-free circuit. For a parity-testable fault, if the primary parity (p_0) differs, the fault is *primary-parity testable*, otherwise it is *secondary-parity testable*. A fault is said to be *parity untestable* if it is not parity testable.

It is well known that [6], [4] if a circuit N realizes a function whose primary parity (p_0) is odd, all stuck-at faults involving any primary input lines of N will be primary-parity testable. Recently Akers [4] has shown that if $p_0 = 0$, but if any subparity $p_i = 1$, then all input stuck-at faults except those where input i is involved can also be detected, and therefore, if $p_0 = 0$ and at least two of the subparities are 1, then complete coverage of input stuck-at faults is again ensured. Clearly, the PBS consisting of all 0's is totally useless as regards to input faults, and the percentile fault coverage will drop very nearly to zero.

From Theorems 1 and 2 one can see that the all-zero signature can occur frequently for supergate-decomposable circuits, calling into question the effectiveness of the parity-bit signature unless the circuits are modified so that they are parity testable.

Definition: A maximal supergate SG(x) in a single-output network N is called *noninternal*, if at least one input line of SG(x) is a primary input of N, otherwise SG(x) is said to be internal.

Theorem 3: Let N be a single-output combinational network N. Then a stuck-at fault f (not necessarily an input fault) in N is parity testable at the primary output of N if and only if condition A or B is satisfied.

Condition A

i) fault f is parity testable in the supergate SG(i) which involves f *and* ii) primary parities of all other maximal supergates are odd.

Condition B

i) fault f is primary-parity testable in SG(i) *and* ii) there exists exactly one even-parity noninternal supergate SG(j), [other than SG(i)] with an input x such that x is a primary input of N and $p_x(F_j) = 1$.

Proof: The proof follows easily from Theorems 1 and 2 and is omitted here for brevity..

Corollary 2: Even one even-parity internal (maximal) supergate [say SG(x)] causes the parity-bit signature of N to be *all-zero* and consequently all stuck-at faults (single or multiple) in $N - \mathrm{SG}(x)$ will be parity untestable. Only some faults in SG(x) might be testable at the primary output. If there are two even-parity noninternal maximal supergates, then the parity-bit signature will again be all-zero, and all stuck-at faults (single or multiple) involving at least one primary input of N will be parity untestable in N.

Remark. One might wonder at this point whether circuits which have all-zero signatures are more likely to occur in practice than those which do not. This is not true if circuits are assumed to implement randomly chosen functions. From statistical arguments it can be shown in such a case that the PBS will be uniformly distributed [4].

IV. Parity-Testable Design

The procedure for making a design parity testable involves three steps: 1) computation of the primary parity of the circuit, 2) logic modification at the supergate level, and 3) parity testable design for the overall circuit. We will illustrate the procedure using the circuit shown in Fig. 2. The circuit is first decomposed into maximal supergates SG(a), SG(b), SG(c), SG(d), and SG(h) as shown.

A. Computation of Primary Parity

The first step is to compute the primary parity realized by the circuit. If the primary parity is odd, no circuit modification is necessary. From Theorem 1, we note that primary parities need to be computed only at the maximal supergate level and the global primary parity can be determined by ANDing the individual parities. Here we describe a method for parity computation based on the binary-decision-diagram representation of the function. Another procedure, based on the circuit topology, is described in [7].

Boolean functions can be represented by a binary decision diagram (BDD) [8]–[10] which can be traversed sequentially for functional evaluation. Several variations of the basic BDD scheme exist in the literature; here we conform to the scheme used in [9] and [10]. We assume that the functionalities of maximal supergates are already available as BDD's. Each nonleaf node x_i in a BDD represents a literal x_i of the function, and it has two outgoing edges for $x_i = 0$ and $x_i = 1$. There are two leaf nodes corresponding to the given function F and its complement $\bar{F}$. A conjunction of Boolean literals appearing in a directed path from the root node to the leaf node F represents an implicant of F. We also assume that all node variables are literals.

Consider, for example, the function realized by supergate SG(h) (Fig. 2), with respect to its own inputs:

$$F_h = c\bar{b}\bar{d} + cad + c\bar{a}b + \bar{c}ab\bar{d} + \bar{a}\bar{b}\bar{c}d\,. \tag{2}$$

The decision diagram for F_h is shown in Fig. 3.

A directed path from the root node to a leaf in a BDD is said to be *complete* if it consists of all n literals in an n-variable function. If the path is not complete (has fewer than n literals) then the corresponding implicant covers an even number of minterms, whereas a

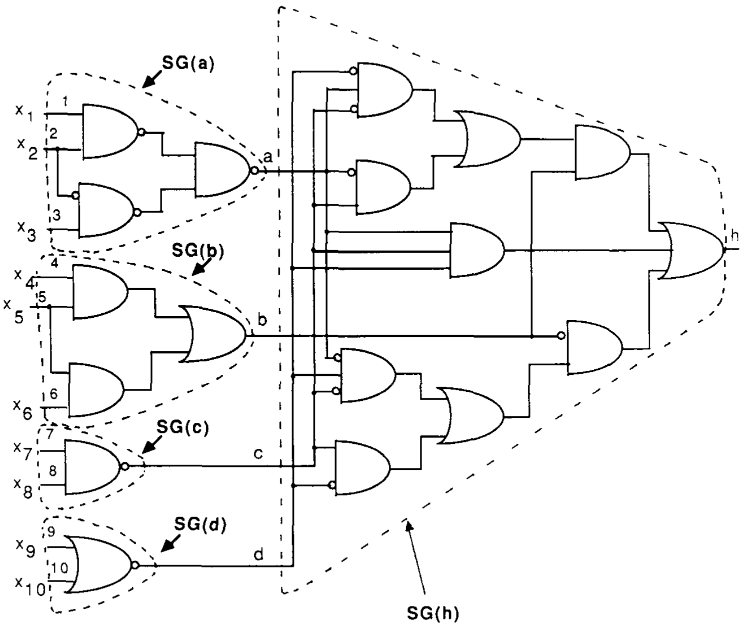

Fig. 2. An illustrative circuit for parity testing.

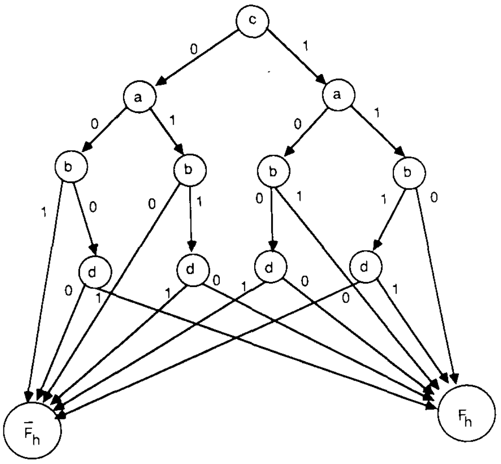

Fig. 3. A binary decision diagram of function F_h.

complete path corresponds to a single minterm. From the definition of a decision diagram, the implicants defined by two distinct paths from the root to a leaf node are disjoint, therefore, the primary parity of the function will be given by

$$p(F) = \{\text{number of complete paths from the root to leaf } F\} \bmod 2.$$

In our example, the diagram has only four complete paths leading to the node F_h. The primary parity of function F_h is therefore zero. Such computation can obviously be performed in time complexity at most $O(e)$, where e denotes the number of edges in the decision diagram, by adopting a standard graph traversal algorithm [11]. The subparities can also be determined in a similar way.

Using the above method, one can compute the primary parities for each supergate individually. In this example, SG(h) is an internal supergate (i.e., none of its inputs is a primary input) with an even primary parity. From Corollary 2, the parity-bit signature for the entire network will be all-zero.

B. Logic Modification of a Supergate

We will now describe a procedure to modify the functionality of each even-parity maximal supergate so as to make its primary parity odd in the test mode. Assume $F(y_1, y_2, \cdots, y_n)$ is an even-parity function denoting the output of such a supergate (in terms of its own inputs). The procedure consists of two steps:

1) Find a maximal subcube[2] of the binary n-cube which covers an odd number of minterms of F and let P be the corresponding product term.

2) Construct a new logic function $F' = F + C \cdot P$ where C is a new literal denoting a control input.

For our example circuit (Fig. 2), only SG(a) and SG(h) have even parities. We illustrate the above procedure for SG(h) which realizes the function F_h given in (2); its Karnaugh map is shown in Fig. 4. A maximal subcube for F_h could be

$$P = b \cdot c \qquad \text{(see Fig. 4).}$$

The augmented function is therefore

$$F_h' = F_h + C \cdot b \cdot c$$

which is realized by network N' as shown in Fig. 5.

Lemma 1: Given a (nonconstant) Boolean function $F(x_1, x_2, \cdots, x_n)$ of n variables, with even primary parity, such a maximal subcube P covering an odd number of minterms of F always exists. Moreover, P will contain at most $(n - 1)$ literals.

Proof: Consider the Karnaugh map of the n-variable function F. Since F is not a constant function, it should have two *adjacent cells* c_1 and c_2 with opposite entries (that is, one and zero). Clearly c_1, c_2 will be a subcube whose product term contains $(n-1)$ literals. This subcube also contains an odd number of minterms of F. If it is not maximal then it must be covered by a bigger subcube with the same property; this subcube would have a fewer number of literals in its product term. ■

Theorem 4: The primary parity of the augmented function F' is odd.

[2] A maximal subcube is one which is not contained within a bigger subcube covering an odd number of minterms in F.

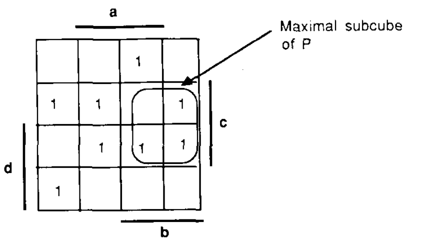

Fig. 4. The Karnaugh map of F_h.

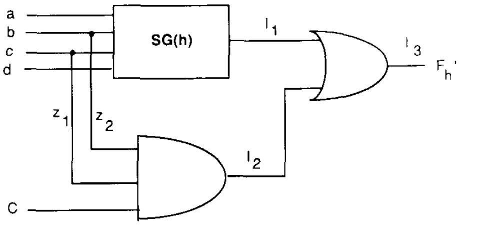

Fig. 5. The augmented circuit implementing F_h'.

Proof: Assume that $\oplus$ denotes modulo-2-sum operation. Then

$$\begin{aligned} p_0(F') &= p_0(F + C \cdot P) \\ &= p_0(F) \oplus p_0(F + P) \\ &= p_0(F) \oplus p_0(F) \oplus p_0(\bar{F}P) \\ &= p_0(\bar{F}P) = 1. \end{aligned}$$

The last equality follows because whenever P (which contains at most $n - 1$ literals) includes an odd number of minterms of F it also includes an odd number of minterms of $\bar{F}$. ■

The hardware overhead for augmenting an even-parity supergate is thus a two-input OR gate, an AND gate with at most $(n - 1)$ inputs, and one control line. The problem of finding a maximal subcube covering an odd number of minterms in an n-variable function F is however complex and may have the worst case complexity $O(2^n)$. This is simply because the number of minterms in F could also be on the order of $O(2^n)$.

A network N is said to be *irredundant* if all stuck-at faults (single or multiple) are detectable, that is, there is some input for which the fault-free function is different from the faulty function. It can be verified that the above augmentation procedure does not introduce new redundancies in the circuit—if N is irredundant so will be N'.

Let us now consider the augmented network N' as shown in Fig. 5.

Theorem 5: a) Any single stuck-at fault or any multiple stuck-at fault involving at least one primary input, or primary output of N', is primary-parity testable at the primary output of N';

b) Any stuck-at fault (single/multiple) in the branch lines $z_1, z_2, \cdots, z_k$ feeding the extra AND gate or in connecting lines l_1, l_2 is also primary parity testable.

Proof: a) easily follows from the fact that $p_0(F') = 1$, and any such fault would make F' vacuous in at least one literal and therefore the primary parity of the faulty function would become even.

b) A fault in line l_1 will also make F' vacuous in at least one literal since the size of the required maximal subcube P never exceeds $(n - 1)$. A fault in l_2 makes F' vacuous in C. As regards branch lines $z_1, z_2, \cdots, z_k$ feeding the AND gate, all stuck-at 0 faults are equivalent to line C stuck-at 0, and hence parity testable. Any stuck-at-1 fault on these lines $z_1, z_2, \cdots$, will also make the primary parity of F' even, since the subcube P is maximal, and therefore any other bigger subcube covering P will include an even number of minterms of F. ■

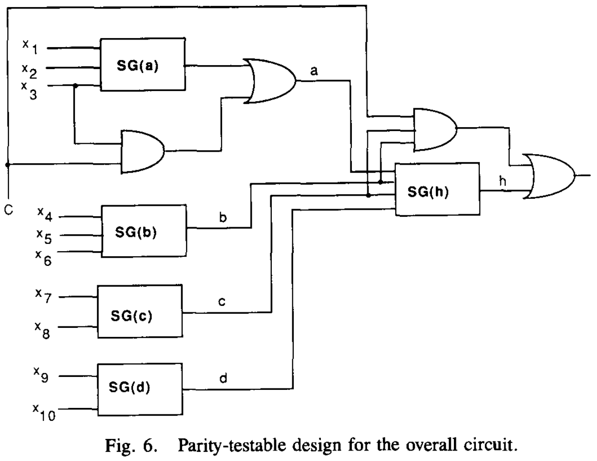

Fig. 6. Parity-testable design for the overall circuit.

C. Parity-Testable Design for the Overall Circuit

We modify each even-parity maximal supergate individually, as described above, to make their primary parity odd. Then we tie all these control lines to form a single control line. The overall testable design for the circuit shown in Fig. 2 is now given in Fig. 6. The method can also be extended to multioutput circuits.

Theorem 6: The primary parity of the global network constructed as above will be odd.

Proof: If we set $C = 0$, the augmented circuit reduces to the original circuit, which realizes an even number of minterms. This happens because the existence of a control line C means there exists at least one supergate SG(x) which required logic modification, and therefore its original primary parity was even. From Theorem 1, the entire original circuit will therefore exhibit even-primary parity when $C = 0$. On the other hand, when C is set to 1, each even-parity supergate SG(x) realizes $(F_x + P_x)$ where F_x is the function realized by SG(x) alone and P_x denotes the maximal subcube required for logic modification of SG(x). The parity of $(F_x + P_x)$ is always odd. The supergates which do not require logic modification already have odd parity. From Theorem 1 again, $C = 1$ contributes an odd parity in the global network. Hence the proof follows. ■

D. Test Procedure and Fault Coverage in the Augmented Circuit

While testing the augmented circuit N', we observe the parity at the primary output *both* for $C = 0$ and $C = 1$. The former corresponds to the subparity p_C, and the combined effect for $C = 0$ and $C = 1$ determines the primary parity of N'.

Since the primary parity of the augmented circuit (N') is now odd, all single/multiple stuck-at faults involving at least one primary input will be parity testable. Moreover, it is easy to show that faults in the extra-logic part involving control line C and faults involving individual input/output lines of all supergates are also primary parity testable. Since we need only one control input, the overall test length is increased by a factor of 2. Note that all faults which were primary-parity testable in the original even-parity circuit N (before augmentation) are still testable in N' when the subparity p_C is observed.

As an example, consider the even-parity function $A \oplus B$ whose testable realization with a control C is shown in Fig. 7. By Theorem 6 all s-a-1 and s-a-0 faults on lines 1, 2, 3, 4, 9, 12, and 13 are primary parity testable. The s-a-1 faults on lines 10 and 11 are equivalent to the s-a-1 fault on line 12. The s-a-0 faults on lines 10 and 11 as well as both s-a-1 and s-a-0 faults on the internal lines 5, 6, 7, and 8 change the primary parity in the original circuit from even to odd. In the augmented circuit, these will be tested when the subparity p_c

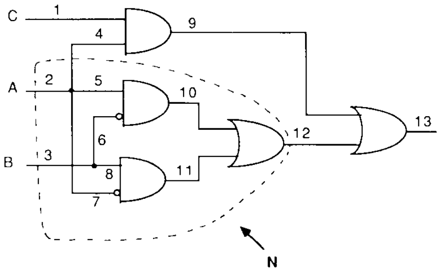

Fig. 7. A testable realization for $A \oplus B$.

is observed. Hence, *all* single and multiple faults are parity testable in N'.

The effectiveness of parity testing, i.e., which faults other than the input and output line faults are sensitive to it, cannot be completely ascertained without knowing the circuit implementation [4], [6]. The testability of faults on internal lines can be partially deduced from Theorem 4. Further evidence and arguments in support of the robustness of parity testing may be found in the two references just cited. Extending Carter's argument [6], it is easy to prove that in a circuit having n primary inputs and k maximal supergates the fault coverage (FC) for multiple faults in the augmented circuit, on the basis of observing primary parity alone, is given by

$$\mathrm{FC} \geq 1 - \frac{1}{3^{n+k}}$$

which asymptotically approaches 100 percent as $(n + k)$ becomes large. We must note, however, that since this result is based on purely combinatorial arguments, it does not guarantee equally high coverage of faults of low multiplicities.

References

[1] S. C. Seth, L. Pan, and V. D. Agrawal, "Predict-probabilistic estimation of digital circuit testability," in *Dig. Papers, FTCS-15*, June 1985, pp. 220-225.

[2] S. C. Seth, B. B. Bhattacharya, and V. D. Agrawal, "An exact analysis for efficient computation of random pattern testability in combinational circuits," in *Dig. Papers, FTCS-16*, Vienna, Austria, 1986, pp. 318-323.

[3] A. Tzidon, I. Berger, and M. Yoeli, "A practical approach to fault detection in combinational networks," *IEEE Trans. Comput.*, vol. C-27, no. 10, pp. 968-971, Oct. 1978.

[4] S. B. Akers, "A parity bit signature for exhaustive testing," in *Proc. Int. Test Conf.*, Sept. 1986, pp. 48-53.

[5] V. H. Tokmen, "Disjoint decomposability of multi-valued functions by spectral means," in *Proc. IEEE Int. Symp. Multiple-Valued Logic*, 1980, pp. 88-93.

[6] W. C. Carter, "The ubiquitous parity bit," in *Proc. 12th Int. Symp. Fault Tolerant Comput. (FTCS-12), Dig. Papers*, June 1982, pp. 289-296.

[7] B. B. Bhattacharya and S. C. Seth, "On reconvergent structure of combinational circuits with applications to parity and syndrome testing," in *Dig. Papers, FTCS-17*, July 1987, pp. 264-269.

[8] C. Y. Lee, "Representation of switching circuits by binary decision programs," *Bell Syst. Tech. J.*, vol. 38, pp. 985-999, July 1959.

[9] S. B. Akers, "Binary decision diagrams," *IEEE Trans. Comput.*, pp. 75-82, June 1978.

[10] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, pp. 677-691, Aug. 1986.

[11] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.

A Note Extending the Analysis of Two-Head Disk Systems to More General Seek-Time Characteristics

A. R. CALDERBANK, E. G. COFFMAN, JR., AND LEOPOLD FLATTO

Abstract—**We analyze a model of a movable-head disk system with two read/write heads maintained a fixed distance d apart on each arm. Successive request-addresses are assumed to be independent random variables, uniformly distributed over the set of cylinders. The purpose of an earlier analysis was to find that value of d which minimizes the expected seek time per request, assuming that seek time varies linearly with the distance z traveled by the heads. In this note, we extend this analysis to more general seek-time characteristics which take into account nonlinear acceleration effects. Detailed results, combining both analysis and simulation experiments, are presented for seek times linear in z^α, $0 \leq \alpha \leq 1$. An unexpected result of the study was that the value of d which minimizes expected seek time is very nearly independent of α.**

Index Terms—**Auxiliary storage systems, disk performance analysis, disk seek time measurement, secondary storage devices, two-head disk systems.**

I. Introduction

We consider a mathematical model of computer disk storage devices having two movable read/write heads. The model approximates the set of storage addresses by the continuous interval [0, 1]. Arriving requests for information at points on this interval are served in first-come-first-served order, and they are modeled by a sequence of independent random variables uniformly distributed over [0, 1]. Serving a request consists first of selecting the head to perform the service, then positioning this head at the requested location, and finally, carrying out the read/write operation itself.

A number of two-head disk systems have been studied within this model; in [2], the reader can find a brief survey and appropriate references. Here, as in [1] and [4], we analyze the most economical system, in which the two heads over each disk surface are situated on the same arm, i.e., the two heads are in a fixed relative position a distance d apart. Thus, when serving a request, both heads must move. One is moved to the requested location, and the other is moved simultaneously an equal distance in the same direction. Both heads must always remain within [0, 1]. Therefore, coverage of the entire interval by the two heads requires that $0 \leq d \leq 1/2$. Moreover, it is clear that the left-head is restricted to the interval $[0, 1-d]$ while the right-head is restricted to $[d, 1]$.

The head selection policy most often studied is the simple nearer-server (NS) rule: for requests in $[d, 1-d]$ the nearer of the two heads is selected to serve the request. Since the two heads must always be kept on the disk surface, requests in $[0, d)$ and $(1-d, 1]$ are always served by the left and right heads, respectively. The objective of the analysis is the steady-state expected time required by the head motion (i.e., the seek time) in serving a request, expressed as a function of the design parameter d.

Seek time is normally taken to be some convex function $C(z)$ of the distance moved. A discussion of seek-time characteristics can be found in the text by Matick [3]. In [1], a constant head speed was normalized to 1, and the seek time was chosen to be just the distance moved, $C(z) = z$; however, the analysis is trivially extended to any linear function $C(z) = az + b$. Note that the constant b is a simplistic model of an initial transient created by acceleration effects.

Manuscript received June 11, 1987; revised October 15, 1988.
The authors are with AT&T Bell Laboratories, Murray Hill, NJ 07974.
IEEE Log Number 8930816.

0018-9340/89/1100-1584$0.100 © 1989 IEEE