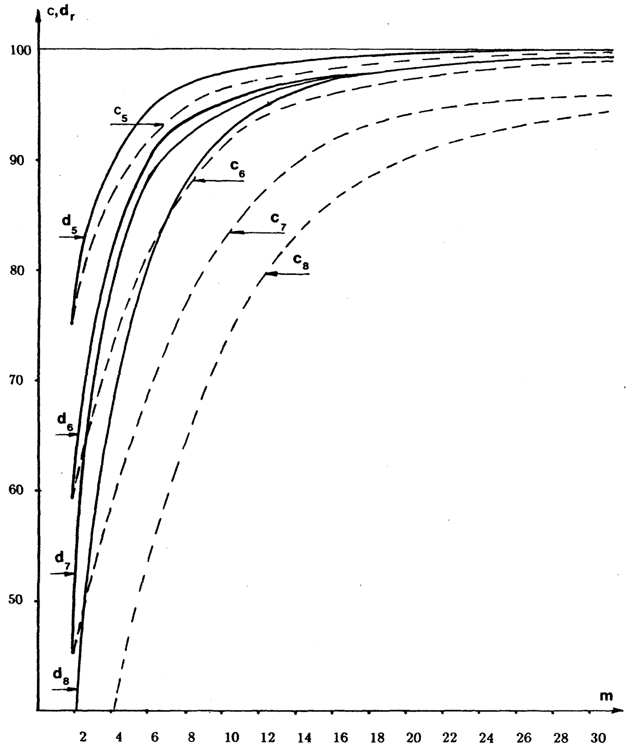

Fig. 7. Multiple fault coverage ratio $dr$ of $T_c$.

In the three above approximations the relative error does not exceed $m2^{-s}$, $2^{-s}/m$ and $2^{s(2-m)}$, respectively. This results in a relative cumulative error in the last expression of the order $m2^{-s}$, which for $m = 20$ and $s = 20$ is less than 0.002 percent. That means that the coverage ratio $d$ of all multiple contact faults by a complete single fault detection test set $T_c$ in an irredundant PLA is approximately

$$d \approx m2^{-s}.$$

## IV. Conclusion

In this correspondence we have investigated the masking phenomenon of contact faults in irredundant PLA's and its influence on the capability of a complete single test set to detect multiple contact faults. As a result of these studies it was shown that any test set $T_c$ detects at least 25 percent of multiple faults of size 5 containing the four-way masking cycle, and, respectively, 37, 52, and 45 percent of faults of size 6, 7, and 8. The ratio increases with $m$, and as the number of rows goes to infinity the ratio of multiple contact faults with four-way masking cycle detected by $T_c$ of size 5, 6, 7, and 8 reaches 50, 75, 86, and 94 percent, respectively. All these faults were considered as nondetectable by $T_c$. Combining the results of this and the previous paper [10] the more realistic bounds of the multiple coverage capability of a complete single contact fault detection test set were established. The coverage ratio $f_r$ of multiple faults of size $r$ not exceeding 8 reaches in the worst case 99 percent for the number of rows equal to 24 and increases even more for larger PLA's. This worst case analysis is based on the assumption that the number $s$, of used crosspoints in a product line is equal to half the total number of crosspoints, $s = n + p/2$. Then, the number of multiple faults including a four-way masking cycle, i.e., multiple faults which may not be detected by $T_c$, is maximal. In a real but irredundant PLA where $s \leq n + p/2$, the coverage is even better. These results, however, cannot be used for PLA's having redundant points, unless the PLA's are converted to crosspoint irredundant for testing purposes. We may expect that each undetected contact fault decreases the coverage but more work has to be done to find quantitative measures.

## Acknowledgment

The authors would like to thank Dr. V. K. Agarwal and N. Cotsapas of the VLSI Design Laboratory of McGill University for their careful reading of the manuscript and their helpful suggestions.

## References

[1] V. K. Agarwal, "Multiple fault detection in programmable logic arrays," *IEEE Trans. Comput.*, vol. C-29, pp. 518–522, June 1980.

[2] V. K. Agarwal and G. M. Masson, "Generic fault characterizations for table-look-up coverage bounding," *IEEE Trans. Comput.*, vol. C-29, pp. 288–299, Apr. 1980.

[3] V. K. Agarwal and A. S. F. Fung, "Multiple fault testing of large circuits by single fault test sets," *IEEE Trans. Comput.*, vol. C-30, pp. 855–865, Nov. 1981.

[4] F. J. O. Dias, "Fault masking in combinational logic circuits," *IEEE Trans. Comput.*, vol. C-24, pp. 476–482, May 1975.

[5] H. Fleisher and L. I. Maissel, "An introduction to array logic," *IBM J. Res. Develop.*, vol. 19, pp. 98–109, Mar. 1975.

[6] M. Fridrich and W. A. Davis, "Minimal fault tests for combinational networks," *IEEE Trans. Comput.*, vol. C-23, pp. 850–859, Aug. 1974.

[7] J. W. Gault, J. P. Robinson, and S. M. Reddy, "Multiple fault detection in combinational networks," *IEEE Trans. Comput.*, vol. C-21, pp. 31–36, Jan. 1972.

[8] D. L. Ostapko and S. J. Hong, "Fault analysis and test generation for programmable logic arrays (PLA's)," *IEEE Trans. Comput.*, vol. C-28, pp. 617–627, Sept. 1979.

[9] J. Rajski and J. Tyszer, "Fault influence function for identification of multiple faults in combinational circuits," *FTCS-13*, Milano, Italy, June 1983, pp. 106–109.

[10] ——, "Combinatorial approach to multiple contact faults coverage in programmable logic arrays," *IEEE Trans. Comput.*, vol. C-34, pp. 549–543, June 1985.

[11] K. S. Ramanatha and N. N. Biswas, "A design for testability of undetectable cross-point faults in programmable logic arrays," *IEEE Trans. Comput.*, vol. C-32, pp. 551–557, June 1983.

[12] J. E. Smith, "Detection of faults in programmable logic arrays," *IEEE Trans. Comput.*, vol. C-28, pp. 845–853, Nov. 1979.

[13] ——, "On necessary and sufficient conditions for multiple fault undetectability," *IEEE Trans. Comput.*, vol. C-28, pp. 801–802, Oct. 1979.

# On the Impossible Class of Faulty Functions in Logic Networks Under Short Circuit Faults

BHARGAB B. BHATTACHARYA AND BIDYUT GUPTA

***Abstract* — The important problem of recognizing *a priori* the class of Boolean functions which are never obtainable from a given combinational network under short circuit faults is almost unexplored, primarily due to lack of understanding of the functional and structural factors that influence the fault behavior in the network. In view of this, a new concept of impossible class of faulty functions (ICFF) is introduced in this correspondence. Several intriguing properties of ICFF are uncovered, namely, the undetectability of input bridging faults, the impossibility of the transformation of a fault free function $F_o$ to a subset or superset of $\bar{F}_o$, and to other functions belonging to the same *P*- and *N*-equivalence classes of $F_o$, etc. The closure amongst the fan-out-free and unate functions under bridging faults is investigated. The impact of ICFF on the testability of the network is also discussed.**

Manuscript received December 5, 1983; revised July 2, 1985.
The authors are with Indian Statistical Institute, Calcutta 700 035, India.
IEEE Log Number 8406291.

***Index Terms*—Bridging faults, combinational logic, fan-out-free functions, fault detection, *N*-equivalence classes, *P*-equivalence classes, short circuit faults, stuck-at faults, unate functions.**

## I. Introduction

With the advent of MOS LSI/VLSI technology, the study of short circuit or bridging faults has received considerable interest in recent times [1]–[10]. The occurrence of such faults is mainly due to the circuit malfunction caused by accidental shorts between two or more conducting paths in the circuit, or due to the breakdown of some insulating medium. In a MOS LSI environment, bridging faults have a substantial probability of occurrence and its physical fault behavior cannot always be modeled by stuck-at faults on its equivalent logic circuit [9], [18].

A bridge fault between two lines $h_1$ and $h_2$ can be logically modeled as either wired-AND or wired-OR function [1]. This fault has the effect of an AND function for positive logic, and that of an OR function for negative logic. In other words, a bridge fault between lines $h_1$ and $h_2$ would change both the functional values $f(h_1)$ and $f(h_2)$ at lines $h_1$ and $h_2$ to $\{f(h_1) \cdot f(h_2)\}$ in case of AND bridging, and to $\{f(h_1) + f(h_2)\}$ in case of OR bridging [1], [2]. In addition, all lines that are directly connected to $h_1$ and $h_2$ in the physical layout of the network would equally be affected [2]. The AND (OR) bridging fault between lines $h_1$ and $h_2$ is usually denoted by $*(h_1/h_2)$ $(+(h_1/h_2))$. Unfortunately, the detection of bridging faults in an arbitrary network is a difficult problem and moreover, some undetectable bridging fault might invalidate a valid stuck-at fault test set [7] which, consequently, makes the test generation approach more complex. Recently a syndrome testable design for easy detection of bridging faults in LSI/VLSI circuits has been reported in [10]. The design philosophy is based on the motivation of devising a network structure in which the set of possible faulty functions is so restricted, that under the assumed fault model, the syndromes of all faulty functions differ from that of the fault free function. Therefore, the awareness of how a short circuit fault can influence the functional behavior of the network, will not only uncover many mysterious properties of faults in a logic circuit but also would be of promising practical importance, in context to design for testability, signature analysis, better characterization of fault-free test data, to name a few. In view of this, the present correspondence introduces the concept of impossible class of faulty functions (ICFF) under short circuit or bridging faults in a logic circuit, and several interesting aspects of fault behavior and their dependence on the network structure are examined in the following sections.

## II. Behavior of Short Circuit Faults in a Logic Network

The study of the impossible class of faulty functions (ICFF) is a challenging problem in digital circuit theory. Since the effect of a fault on the functional behavior of the network is intimately entwined with the network topology, an exploration of ICFF with all of its endless ramifications is rather a complex task. The characterization of ICFF under short circuit faults in particular, is more difficult as compared to its stuck-at fault counterpart. Before presenting our main results, we will introduce some definitions and notations.

*Definition 1:* Given a circuit realization $N$ of a Boolean function $F_o$, the functions which are never obtainable under any fault in $N$ are called *impossible class of faulty functions (ICFF)* of $F_o$ with respect to $N$. To be more particular, ICFF can be defined separately for stuck-at and bridging faults.

*Definition 2:* If in a network $N$ realizing a Boolean function $F_o$, a fault $f$ causes to create a faulty function $F_f$, we symbolize the event as

$$F_o \xrightarrow[N]{f} F_f.$$

*Definition 3:* Two Boolean functions $F_1$ and $F_2$ are said to be in the same $P(N)$-equivalence class [15], if one can be transformed in the other by permitting (negating) the primary input literals.

Some results regarding the behavior of ICFF under stuck-at faults are due to Hayes [11], [12], and Fujiwara [13]. For instance, it is known that no single stuck-at fault in an irredundant network realizing a nonconstant function $F_o$ can change it to $\overline{F}_o$ [11], [12]. The result is also true for multiple faults in two-level irredundant networks and in special class of EX-OR tree networks. Fujiwara [13] presented some closure properties among the possible class of faulty functions, particularly among the unate and fan-out-free functions. Some anomalous behavior of stuck-at faults has recently been reported in [14] where it has been shown that even a single stuck-at fault in an irredundant combinational network can change the fault free function $F_o$ to its $P$-equivalence class, which focuses on a new kind of logical redundancy. We will now investigate whether similar results also hold good for bridge fault model.

### A. ICFF Under Short Circuit or Bridging Faults

Given a Boolean function $F_o$ and its realization $N$, it would be interesting it we could foresee the class of the faulty functions which are never obtainable under any bridge fault in $N$. The following theorem presented below depicts an interesting property of bridging fault in this respect. We assume that the bridging fault does not create a feedback.

*Theorem 1:* Let $N$ be any single-output irredundant combinational network of arbitrary topology realizing a nonconstant function $F_o = f(x_1, x_2, \cdots, x_n)$. Then there does not exist a bridge fault $f_b$ between any two lines in $N$ which changes $F_o$ to a nonconstant faulty function $F_f$ such that

$$F_f = \overline{F}_o, \quad \text{or} \quad F_f \subset \overline{F}_o, \quad \text{or} \quad F_f \supset \overline{F}_o.$$

In other words,

$$\sim \exists F_o \, \exists N \, \exists f_b (F_o \xrightarrow[N]{f_b} F_f), \qquad \text{where } F_f = \overline{F}_o, \quad \text{or} \quad F_f \subset \overline{F}_o, \quad \text{or} \quad F_f \supset \overline{F}_o$$

is always true.

*Proof:* Let us assume that $f_b$ involves two arbitrary lines, say $h$ and $m$ in $N$, and it is of OR type, i.e., $f_b$: $+(h/m)$. Let the functional values of lines $h$ and $m$ be $f(h)$ and $f(m)$, respectively. Note that if any arbitrary line, say $h$, happens to be a fanout stem or fanout branch line having line function $f(h)$, then all lines emanating from the parent stem line have the same line function $f(h)$. Incorporating this fact, we can always express the function $F_o$ in terms of the functional values of lines $h$ and $m$ as follows:

$$F_o = A_1 f(h) f(m) + A_2 \overline{f}(h) f(m) + A_3 f(h) \overline{f}(m) + A_4 \overline{f}(h) \overline{f}(m) + A_5$$

where $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ are Boolean functions of literals $(x_1, x_2, \cdots, x_n)$. Since, $f_b$ is an OR bridging fault, both lines $h$ and $m$ would assume a functional value $\{f(h) + f(m)\}$ under $f_b$. Therefore, the faulty function $F_f$ can be expressed as

$$F_f = A_1 \, (f(h) + f(m)) + A_4 \overline{f}(h) \overline{f}(m) + A_5.$$

Note that since $N$ is irredundant, $f(h)$ and $f(m)$ are nonconstant functions. Now the intersection between the fault free function $F_o$ and faulty function $F_f$ would become

$$F_o \cdot F_f = A_1 f(h) f(m) + A_1 A_2 \overline{f}(h) f(m) + A_1 A_3 f(h) \overline{f}(m) + A_4 \overline{f}(h) \overline{f}(m) + A_5. \tag{1}$$

To prove the theorem by contradiction, we assume that the bridge fault $f_b$ creates a faulty function $F_f$ such that $F_f \subseteq \overline{F}_o$. If it so happens, then one must have $F_o \cdot F_f = 0$, which means that each term of expression (1) must be individually zero. In particular, therefore,

$$A_1 f(h) f(m) = 0, \; A_4 \overline{f}(h) \overline{f}(m) = 0, \text{ and } A_5 = 0$$

which implies that the fault free function $F_o$ must be of the form

$$F_o = A_2 \overline{f}(h) f(m) + A_3 f(h) \overline{f}(m).$$

Therefore, the faulty function $F_f$ under $+(h/m)$ would be

$$F_f = A_2\,\overline{(f(h) + f(m))}\,(f(h) + f(m)) + A_3\,(f(h) + f(m))\,\overline{(f(h) + f(m))} = 0\,.$$

Thus, the faulty function $F_f$ under $f_b$ can be a subset of, or equal to $\overline{F}_o$ if and only if $F_f = 0$, i.e., a null function, which means that so long as $F_f$ is a nonconstant function, $F_f \not\subseteq \overline{F}_o$.

To prove that $F_f \not\supset \overline{F}_o$ under any bridge fault $f_b$, we assume the existence of a network $N$ realizing $F_o$ such that under some bridge fault $f_b$, the faulty function $F_f$ becomes a superset of $\overline{F}_o$, i.e., $F_o \xrightarrow[N]{f_b} F_f$ such that, $F_f \supset \overline{F}_o$. Let us now construct a network $N'$ by adding an inverter at the output of $N$. $N'$ would therefore realize a fault free function $\overline{F}_o$. The occurrence of the same bridge fault $f_b$ in $N'$ would therefore create a faulty function $F_f'$ which is equal to $\overline{F}_f$. Therefore, we have

$$\overline{F}_o \xrightarrow[N']{f_b} F_f' \text{ where } F_f' = \overline{F}_f\,.$$

Now, $\overline{F}_o \cdot F_f' = \overline{F}_o \cdot \overline{F}_f = 0$ since $\overline{F}_o \subset F_f$ by assumption, meaning that in the network $N'$ there exists a bridge fault $f_b$ under which the intersection between the fault free and faulty function is null or in other words, the faulty function is a subset of the complement of the fault free function. This is clearly a contradiction as proved earlier. Therefore, $F_f \not\supset \overline{F}_o$. The case for the AND type bridging fault $*(h/m)$ can be similarly proved. Hence, the theorem follows. Q.E.D.

Note that similar things also happen in case of single stuck-at faults in a nonredundant network [11], [12]. Moreover, for multiple stuck-at faults ($f_m$) in irredundant circuits, no example is yet known [12] where $F_o \xrightarrow[N]{f_m} \overline{F}_o$. In contrast, there exist logic nets in which a multiple bridging fault can change the fault free function $F_o$ to $\overline{F}_o$ even if the circuit realization is irredundant. The following example makes this transparent.

*Example 1:* Consider the network shown in Fig. 1, which realizes the function $F_o = x_1 \oplus x_2$. Clearly it is an irredundant realization of $F_o$. Consider now a multiple bridging fault: $\{+(h_1/h_2), +(h_3/h_4), +(h_5/h_6)\}$. The faulty function becomes $F_f = x_1\,x_2 + \overline{x}_1\,\overline{x}_2 = \overline{F}_o$. Note that, inverters $I_1, I_2, I_3, I_4$ are included in the circuit to inhibit the flow of the logical effects of faults $+(h_1/h_2)$ and $+(h_3/h_4)$ toward the primary stem lines.

### *B. ICFF Under Bridging Faults at Primary Inputs in Logic Circuits*

The detection of input bridging faults, and some of its properties have been discussed in [8]. Here we will concentrate mainly on the problem of undetectability of bridging faults at the primary input lines in a logic circuit. Some other interesting properties of ICFF are also discussed. The logical model of input bridging faults is shown in Fig. 2.

*Theorem 2(a):* Let $N$ be any single-output combinational network realizing a Boolean function $F_o = f(x_1, x_2, \cdots, x_n)$. Then a multiple OR bridging fault $f_{bm}$ involving $s$ input lines $x_1, x_2, \cdots, x_s$, $s \leq n$, is undetectable if and only if $F_o$ can be expressed as

$$F_o = A_1\,(x_1 + x_2 + \cdots + x_s) + A_2\,\overline{x}_1\,\overline{x}_2 \cdots \overline{x}_s + A_3 \qquad (2)$$

where, $A_1$, $A_2$, $A_3$ are Boolean functions independent of literals $x_1, x_2, \cdots, x_s$. In other words, $F_o \in \text{ICFF}(F_o)$, under a multiple OR bridging fault involving primary inputs, whenever (2) does not hold well.

*Theorem 2(b):* A multiple AND bridging fault of multiplicity $s$, involving $s$ input lines in a network $N$ is undetectable, if and only if

$$F_o = A_1'\,(x_1\,x_2 \cdots x_s) + A_2'\,(\overline{x}_1 + \overline{x}_2 + \cdots + \overline{x}_s) + A_3' \qquad (3)$$

where, $A_1'$, $A_2'$, $A_3'$ are functions independent of $x_1, x_2, \cdots, x_s$.

The proof of theorem 2 is trivial and is therefore omitted.

*Corollary 1:* If an OR bridging fault $+(x_1, x_2, \cdots, x_s)$ among $s$ primary inputs is detectable, then all bridging faults of the following types, namely,

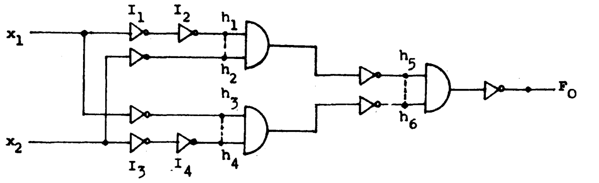

Fig. 1. Network $N$ realizing $F_o = x_1 \oplus x_2$.

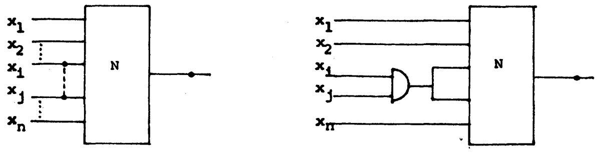

Fig. 2. Logical model of input bridging fault $*(x_i/x_j)$.

$$+(x_1, x_2, \cdots, x_s, x_{s+1}, \cdots, x_q) \text{ or } \{+(x_1, x_2, \cdots, x_s), +(x_{s+1}, x_{s+2}, \cdots, x_r), +(x_{r+1}, x_{r+2}, \cdots, x_m)\} \text{ or } \{+(x_1, x_2, \cdots, x_s, x_{s+1}, \cdots, x_q), +(x_{q+1}, \cdots, x_r), +(x_{r+1}, \cdots, x_m) \cdots\},$$

are detectable. Similar things will also be valid for AND bridging faults.

*Theorem 3:* Let $f_{bm}^{+}$ and $f_{bm}^{*}$ denote multiple OR and AND bridging faults, respectively, involving $s$ input lines in a network $N$. Then if $f_{bm}^{+}$ ($f_{bm}^{*}$) is undetectable, $f_{bm}^{*}$ ($f_{bm}^{+}$) is detectable. In other words,

$$F_o \xrightarrow[N]{f_{bm}^{+}} F_o, \text{ and } F_o \xrightarrow[N]{f_{bm}^{*}} F_o$$

cannot be simultaneously true.

*Proof:* Let $f_{bm}^{+}: +(x_1, x_2, \cdots, x_s)$ and $f_{bm}^{*}: *(x_1, x_2, \cdots, x_s)$. Assume, $F_o \xrightarrow[N]{f_{bm}^{+}} F_o$. Then by theorem 2(a),

$$F_o = A_1\,(x_1 + x_2 + \cdots + x_s) + A_2\,\overline{x}_1\,\overline{x}_2 \cdots \overline{x}_s + A_3\,.$$

Also if $F_o \xrightarrow[N]{f_{bm}^{*}} F_o$, then by theorem 2(b),

$$F_o = A_1'\,x_1\,x_2 \cdots x_s + A_2'\,(\overline{x}_1 + \overline{x}_2 + \cdots + \overline{x}_s) + A_3'$$

where $A$'s and $A''$s are independent of $x_1, x_2, \cdots, x_s$. Clearly, this is a contradiction unless the cardinality of the set $\{1, 2, \cdots, s\}$ is 1, thereby completing the theorem. Q.E.D.

*Corollary 2:* If an OR (AND) bridging fault of multiplicity $s \geq 2$ at the primary inputs in $N$ is undetectable, then any AND (OR) primary input bridging fault of multiplicity $q \leq s$, ($q \geq 2$) is detectable.

It may be noted that the AND (OR) bridging fault model is appropriate for a positive (negative) logic environment. Therefore, corollary 2 has the following interpretation: the presence of an undetectable input bridging fault in a network realized with positive logic implies detectability of the same when the same network topology is realized with negative logic, and vice-versa.

The above theorems describing properties of undetectable input bridging faults are generalized versions of those presented in [8]. We will now investigate some other properties of ICFF, namely the possibility of changing the fault free function to some other functions belonging to the same $P$ or $N$-equivalence class under bridging faults at primary inputs. In this context it may be noted that even a single stuck-at fault in a nonredundant network can change the fault free function to its $P$ or $N$-equivalence class [14], whereas for stuck-at faults at primary inputs and in special class of networks, such transformation is inhibited.

*Theorem 4:* Let $N'$ be an arbitrary $n$ input, single-output network realizing a nonvacuous $n$ variable Boolean function $F_o = f(x_1, x_2, \cdots, x_i, x_j, \cdots, x_n)$. Then any AND (OR) bridging fault between two primary inputs $|x_i, x_j|$ cannot change the function $F_o$ to its $N$ equivalence class with respect to $x_i$, $x_j$ or both, i.e., $F_o$ cannot be transformed to the faulty function by negating the variable $x_i$ or $x_j$ or both.

*Proof:* Without any loss of generality we consider an AND bridging fault $f_b$ between two primary lines $x_i$, $x_j$, i.e., $f_b{:}*(x_i/x_j)$. We can always express $F_o$ as

$$F_o = A_1 x_i x_j + A_2 \bar{x}_i x_j + A_3 x_i \bar{x}_j + A_4 \bar{x}_i \bar{x}_j + A_5 \tag{4}$$

where, $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ are Boolean functions independent of $x_i$, $x_j$. Under fault $*(x_i/x_j)$, the faulty function $F_f$ would become

$$F_f = A_1 x_i x_j + A_4 (\bar{x}_i + \bar{x}_j) + A_5 . \tag{5}$$

Assume without any loss of generality, that $F_f$ is in $N$-equivalence class of $F_o$ with respect to the literal $x_i$, i.e., $F_o \xrightarrow[N']{f_b} F_f$, such that $F_f \in N(F_o)$. Changing $x_i$ to $\bar{x}_i$ and vice-versa in (5) and equating it to (4), we therefore get the following equality:

$$\begin{aligned}(A_1 x_j + A_3 \bar{x}_j + A_5)\, x_i + (A_2 x_j + A_4 \bar{x}_j + A_5)\, \bar{x}_i& \\ = (A_4 + A_4 \bar{x}_i + A_5)\, x_i + (A_1 x_j + A_4 \bar{x}_j + A_5)\, \bar{x}_i&.\end{aligned}$$

A little simplification will now yield

$$\begin{aligned}x_i\, [(A_1 x_j + A_3 \bar{x}_j + A_5) \oplus (A_4 + A_5)]& \\ = \bar{x}_i\, [(A_2 x_j + A_4 \bar{x}_j + A_5) \oplus (A_1 x_j + A_4 \bar{x}_j + A_5)]&.\end{aligned}$$

Since all $A$'s are independent of $x_i, x_j$, the equality can be satisfied if and only if both sides are individually zero, implying that

$$A_1 x_j + A_3 \bar{x}_j + A_5 = A_4 + A_5 \tag{6}$$

and

$$A_2 x_j + A_4 \bar{x}_j + A_5 = A_1 x_j + A_4 \bar{x}_j + A_5 . \tag{7}$$

From relation (6) it is easy to observe that

$$\bar{A}_5\, [A_1 x_j \oplus A_3 \bar{x}_j] = \bar{A}_5 A_4$$

or

$$\bar{A}_5\, [A_1 x_j \oplus A_3 \bar{x}_j \oplus A_4] = 0$$

or

$$\bar{A}_5\, [(A_1 \oplus A_4)\, x_j \oplus (A_3 \oplus A_4)\, \bar{x}_j] = 0 .$$

Therefore, $\bar{A}_5\, (A_1 \oplus A_4)\, x_j = \bar{A}_5\, (A_3 \oplus A_4)\, \bar{x}_j$. By similar reasoning, this can happen only if both sides of above relation are individually zero, i.e.,

$$\bar{A}_5\, [A_1 \oplus A_4] = 0 = \bar{A}_5\, [A_3 \oplus A_4]$$

implying that

$$\begin{aligned}A_1 \oplus A_4 &\subseteq A_5 \\ A_3 \oplus A_4 &\subseteq A_5 .\end{aligned} \tag{8}$$

Similarly, from relation (7) it is evident that

$$\bar{A}_5\, A_2\, x_j = \bar{A}_5\, A_1\, x_j .$$

In other words, $\bar{A}_5\, [A_1 \oplus A_2]\, x_j = 0$. Because of the independence of $A$'s on $x_j$ this implies

$$A_1 \oplus A_2 \subseteq A_5 . \tag{9}$$

Note that, if $X, Y, Z$ are Boolean functions such that, $X \subseteq Z$ and $Y \subseteq Z$, then $X \oplus Y \subseteq Z$. Applications of this idea on relations (8) and (9) therefore yields the following series of logical implications:

$$\left.\begin{aligned}A_1 \oplus A_4 &\subseteq A_5 \\ A_3 \oplus A_4 &\subseteq A_5 \\ A_1 \oplus A_2 &\subseteq A_5 \\ A_1 \oplus A_3 &\subseteq A_5 \\ A_2 \oplus A_4 &\subseteq A_5 \\ A_2 \oplus A_3 &\subseteq A_5 .\end{aligned}\right\} \tag{10}$$

From (10) it is easy to observe that

$$A_5 \supseteq (A_1 + A_2 + A_3 + A_4)(\bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \bar{A}_4) . \tag{11}$$

We will now consider three cases separately.

*Case 1:* $F_o$ is such a function that $A_5 = 0$. Then from (10) we obtain, $A_1 = A_2 = A_3 = A_4$, which means that $F_o = A_1$ [from (4)], thereby contradicting the hypothesis that $F_o$ is nonvacuous in $(x_i, x_j)$.

*Case 2:* $A_5 \neq 0$ and any one of four functions $A_1, A_2, A_3, A_4$ is 0. Let $A_1 = 0$. Then, from (10), $A_2 \subseteq A_5; A_3 \subseteq A_5; A_4 \subseteq A_5$. Therefore, from (4), $F_o = A_5$, again contradicting the nonvacuousness of $F_o$.

*Case 3:* $A_1 \neq 0, A_2 \neq 0, A_3 \neq 0, A_4 \neq 0, A_5 \neq 0$. Since, $A_5 \supseteq (A_1 + A_2 + A_3 + A_4)(\bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \bar{A}_4)$ from (11), $F_o$ can be expressed as [from (4)] follows:

$$\begin{aligned}F_o &= A_1 A_2 A_3 A_4\, x_i x_j + A_1 A_2 A_3 A_4\, \bar{x}_i x_j \\ &\quad + A_1 A_2 A_3 A_4\, x_i \bar{x}_j + A_1 A_2 A_3 A_4\, \bar{x}_i \bar{x}_j + A_5 \\ &= A_1 A_2 A_3 A_4 + A_5\end{aligned}$$

implying that $F_o$ is vacuous in $x_i, x_j$. The case that $F_f$ cannot be transformed to $F_o$ by negating both $x_i, x_j$ can be similarly shown, thereby completing the proof. Q.E.D.

*Theorem 5:* Let $N'$ be any arbitrary $n$-input, single-output network realizing an $n$ variable nonvacuous function $F_o = f(x_1, x_2, \cdots, x_i, x_j, x_k, \cdots, x_n)$. Then no AND (OR) bridging between any two primary inputs say, $(x_i, x_j)$, can change $F_o$ to a faulty function $F_f$ where $F_f$ can be transformed to $F_o$ by negating a third literal $x_k$, i.e.,

$$F_o \xrightarrow[N']{\overset{*}{+}(x_i/x_j)} F_f$$

where $F_f \in N\ (F_o)$, with respect to negation of literal $x_k$, can never be true.

*Proof:* Let $F_o$ be expressed as follows:

$$\begin{aligned}F_o = {}& A_1 x_i x_j x_k + A_2 \bar{x}_i x_j x_k + A_3 x_i \bar{x}_j x_k + A_4 \bar{x}_i \bar{x}_j x_k \\ & + A_1' x_i x_j \bar{x}_k + A_2' \bar{x}_i x_j \bar{x}_k + A_3' x_i \bar{x}_j \bar{x}_k + A_4' \bar{x}_i \bar{x}_j \bar{x}_k + A_5\end{aligned}$$

where, all $A$'s and $A'$'s are independent of $x_i, x_j, x_k$. Assume that a fault $f_b{:}*(x_i/x_j)$ occurs at primary inputs $x_i, x_j$ of $N'$. The faulty function $F_f$ is therefore

$$\begin{aligned}F_f = {}& A_1 x_i x_j x_k + A_4(\bar{x}_i + \bar{x}_j)\, x_k \\ & + A_1' x_i x_j \bar{x}_k + A_4'(\bar{x}_i + \bar{x}_j)\bar{x}_k + A_5 .\end{aligned}$$

If $F_o \xrightarrow[N']{*(x_i/x_j)} F_f$ where $F_f$ lies in $N$-equivalence class of $F_o$, with respect to $x_k$, then negating literal $x_k$ in $F_f$, we can equate it to $F_o$. It is not difficult to see that such an equality implies that

$$\begin{aligned}&A_1 \oplus A_1' \subseteq A_5, \quad && A_2' \oplus A_4 \subseteq A_5, \quad && A_3' \oplus A_4 \subseteq A_5 \\ &A_4 \oplus A_4' \subseteq A_5, && A_2 \oplus A_4' \subseteq A_5, && A_3 \oplus A_4' \subseteq A_5\end{aligned}$$

from which we get, $A_2 \oplus A_2' \subseteq A_5$, $A_3 \oplus A_3' \subseteq A_5$.

Using these relations one can easily conclude that $F_o$ is independent of $x_k$, which contradicts the nonvacuousness of $F_o$, thereby proving the theorem. Q.E.D.

Based on the same proof technique as that used in proving Theorems 4 and 5, we can claim the following general result regarding the effect of a bridging fault at primary inputs of the network.

*Theorem 6:* Let $N'$ be any arbitrary network realizing an $n$ variable nonvacuous Boolean function $F_o$. Then there does not exist any AND (OR) bridging fault (excluding the undetectable faults) involving primary inputs of $N'$ which can change $F_o$ to its $N$- or $P$-equivalence class. In other words, for all detectable faults $f_b$'s at primary inputs,

$$\{F_f \mid F_f \in P(F_o), F_f \in N(F_o)\} \subseteq \text{ICFF}(F_o)$$

in all networks having irredundant primary input leads.

### C. Further Study of ICFF Under Bridge Faults

The general problem of determining under what conditions there exists an irredundant network in which any arbitrary bridge fault $f_b$ can change the fault free function to its $P$- or $N$-equivalence class is still an open problem. However, for special classes of bridging faults, not necessarily at the input level, one can present the following theorem.

*Definition 4:* Let $h$ be any path from a line in a network to the output and let $N_h$ denote the number of inversions along $h$. Then the parity of path $h$, denoted by $p_h$ is defined as $N_h$ (mod 2).

*Theorem 7:* Let $h$ and $k$ be any two lines in a single-output combinational network $N'$ realizing a nonvacuous Boolean function $F_o$ of $n$ variables. If all paths emanating from $h$ to the network output (if $h$ happens to be a fanout stem or a fanout branch line directly connected to the stem (i.e., without any inverter in between) then the path list also includes those emanating from all branch lines of the parent stem line) are of equal parity, say $p_h$ and similarly if all paths from line $k$ to the network output are of equal parity, say $p_k$ and if $p_h = p_k$, then no AND (OR) bridging fault between $h$ and $k$, if it is detectable, can change the fault free function $F_o$ to its $P$- or $N$-equivalence class.

*Proof:* Let $f(h)$ and $f(k)$ denote functional values at lines $h$ and $k$, respectively. Without any loss of generality, we assume $p_h = p_k = 0$. We can now express the fault free function as follows:

$$F_o = A_1 f(h) + A_2 f(k) + A_3 f(h) f(k) + A_4$$

where $A_1, A_2, A_3, A_4$ are Boolean functions of input literals but are independent of $f(h)$ and $f(k)$.

Let the fault be $f_b: +(h/k)$. The faulty function $F_f$ is therefore given by

$$F_f = A_1\,(f(h) + f(k)) + A_2\,(f(h) + f(k)) + A_3\,(f(h) + f(k)) + A_4.$$

Clearly, $F_f \supseteq F_o$, and for detectable faults $F_f \neq F_o$. Therefore, $F_f \supset F_o$, implying that $F_f \notin P(F_o)$ and $F_f \notin N(F_o)$. For AND type bridging similar reasoning will do, thereby completing the theorem.

Q.E.D.

Like ICFF under stuck-at faults, we have characterized several impossible class of faulty functions under bridging or short circuit faults in a logic circuit. In [13], Fujiwara described some closure properties among fan-out free and unate functions under stuck-at faults. In other words, in a tree network, all possible faulty functions will be fan-out-free functions and in a network realizing a unate function, in which the number of inversions along any path connecting two points in the circuit, are the same (unate composition), all faulty functions would be unate functions when stuck-at faults occur. In case of bridging faults however, the effect of the fault is to introduce a fanout point in the network [Fig. 2] and therefore the closure with respect to unateness and fan-out-freedom, is in general, not preserved. But for a restricted class of short circuit faults this closure property can still be maintained. In view of this we present the following observations.

*Lemma 1:* Let $N$ be any two-level AND-OR or OR-AND tree network (with inverters if any, only at the primary inputs) realizing a fan-out-free function $F_o$. Let $f_b$ be a nonfeedback bridge fault in between lines $h$ and $m$ in $N$, such that neither $h$ nor $m$ is an input to any inverter. Then the faulty function $F_f$ under $f_b$ is a fan-out-free function.

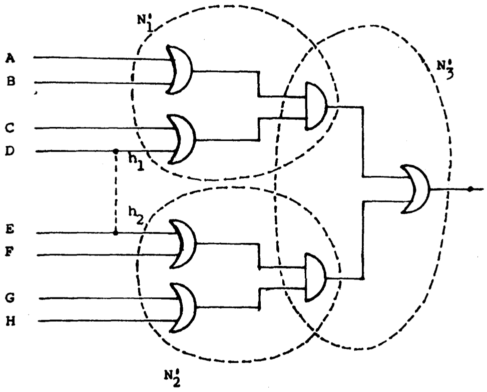

Fig. 3. Network $N$ with subcircuits $N'_1, N'_2, N'_3$.

The proof of lemma 1 is simple and is omitted. As an immediate consequence of this we now get the following theorem.

*Theorem 8:* Let $F_o$ be an arbitrary fan-out-free function realized by a multilevel tree network $N$ composed of AND/OR gates with inverters, if any, only at the input level. Assume that the inverter-inputs are not vulnerable to any circuit fault. Let $f_b$ be a bridging fault involving two lines $h$ and $m$, such that $h$ and $m$ belong to a two-level AND-OR or OR-AND subcircuit $N'$ of $N$. Then the fault function $F_f$ is a fan-out-free function.

*Example 2:* Consider the network $N$ shown in Fig. 3, realizing a fan-out-free function $F_o = (A + B) \cdot (C + D) + (E + F) \cdot (G + H)$. The embedded two-level subcircuits $N'_1, N'_2, N'_3$ in $N$ are shown within dotted lines. Any bridging fault which is confined within any of these subcircuits will result a fan-out-free faulty function. For a bridging fault $*(h_1/h_2)$ in $N$, the faulty function would be $F_f = (A + B) \cdot (C + DE) + (DE + F) \cdot (G + H)$, which however, is not a fan-out-free function. Note that fault $*(h_1/h_2)$ does not belong to any two-level subcircuit of $N$.

It is well known that any fan-out-free function can be realized by a multilevel tree network with only AND-OR gates with inverters at the input level. Theorem 8 illustrates the fact that, albeit the effect of a bridging fault is logically equivalent to creation of a fanout point in the circuit, the class of fan-out-free functions satisfies the closure property with respect to bridging faults confined within all two-level subcircuits in a tree network. The next theorem presents similar properties regarding the closure amongst unate functions.

*Theorem 9:* Let $N$ be a network realizing a unate function $F_o$ such that number of inversions in any path connecting two points in $N$ are same (unate composition). Let $f_b$ denote a bridging fault between two lines $h$ and $m$ such that $p_h = p_m$. Then the faulty function $F_f$ must be a unate function.

## III. Conclusion

In this correspondence, several interesting properties reflecting the behavior of short circuit faults in logic networks are explored. There exist however, many other complex situations, which require further investigation in order that ICFF under short circuit faults can be more well-defined in an arbitrary circuit. Needless to say, the cardinality of ICFF would be a potential candidate for providing a measure of the testability of the network. This happens because, when exhaustive testing methodologies like syndrome testing [16], testing by verifying Rademacher–Walsh transformations [17], etc., are used for implementing built-in-tests (BIT), the presence of a larger set of ICFF in a network will constrict the possible set of faulty functions, and consequently the test data, i.e., the information about the fault free circuit need not have to distinguish the fault free function from each of the remaining $(2^{2^n} - 1)$ functions for an $n$-input circuit. In view of LSI/VLSI technology, the difficult problem of generating test sets for detecting bridging faults can be circumvented by resorting to a syndrome testable design which takes care of detecting short circuit faults as well [10]. Better comprehension of ICFF and its application to testable design with an objective of having higher fault coverage in LSI/VLSI circuits would be a challenging area of future research.

## References

[1] K. C. Y. Mei, "Bridging and stuck-at faults," *IEEE Trans. Comput.*, vol. C-23, pp. 720–727, July 1974.

[2] A. D. Friedman, "Diagnosis of short circuit faults in combinational circuits," *IEEE Trans. Comput.*, vol. C-23, pp. 746–752, July 1974.

[3] S. Xu and S. Y. H. Su, "Testing feedback bridging faults among internal, input and output lines by two patterns," in *Proc. IEEE Int. Conf. Circuits Comput., ICCC*, 1982, pp. 214–217.

[4] M. Abramovici and P. R. Menon, "A practical approach to fault simulation and test generation for bridging faults," in *Proc. IEEE Int. Test Conf.*, 1983, pp. 138–142.

[5] P. Lamoureux and V. K. Agrawal, "Nonstuck-at fault detection in *n*MOS circuits by region analysis," in *Proc. IEEE Int. Test Conf.*, 1983, pp. 129–137.

[6] A. Isoupvicz, "Optimal detection of bridge faults and stuck-at faults in two-level logic," *IEEE Trans. Comput.*, vol. C-27, pp. 452–455, May 1978.

[7] K. L. Kodandapani and D. K. Pradhan, "Undetectability of bridging faults and validity of stuck-at fault tests," *IEEE Trans. Comput.*, vol. C-29, pp. 55–59, Jan. 1980.

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

[9] J. Galiay *et al.*, "Physical versus logical fault models in MOS LSI circuits: impact on their testability," *IEEE Trans. Comput.*, vol. C-29, pp. 527–531, June 1980.

[10] B. B. Bhattacharya and B. Gupta, "Syndrome testable design of combinational networks for detecting stuck-at and bridging faults," in *Proc. IEEE Int. Test Conf.*, 1983, pp. 446–452.

[11] J. P. Hayes, "On realizations of Boolean functions requiring a minimal number of tests," *IEEE Trans. Comput.*, vol. C-20, pp. 1506–1513, Dec. 1971.

[12] ——, "Transition count testing of combinational circuits," *IEEE Trans. Comput.*, vol. C-25, pp. 613–620, June 1976.

[13] H. Fujiwara, "On closedness and test complexity of logic circuits," *IEEE Trans. Comput.*, vol. C-30, pp. 556–562, Aug. 1981.

[14] B. B. Bhattacharya and B. Gupta, "Anomalous effect of a stuck-at fault in a combinational logic circuit," *Proc. IEEE*, vol. 71, pp. 779–780, June 1983.

[15] S. L. Hurst, *Logical Processing of Digital Signals*. New York: Crane Russak, 1978.

[16] J. Savir, "Syndrome testable design of combinational circuits," *IEEE Trans. Comput.*, vol. C-29, pp. 442–457, June 1980.

[17] A. K. Susskind, "Testing by verifying Walsh coefficients," in *Proc. FTCS-11*, June 1981, pp. 206–208.

[18] B. B. Bhattacharya and B. Gupta, "Logical modeling of physical failures and their inherent syndrome testability in MOS LSI/VLSI networks," in *Proc. IEEE Int. Test Conf.*, 1984, pp. 847–855.

# On System Diagnosability in the Presence of Hybrid Faults

A. SENGUPTA, A. SEN, AND S. BANDYOPADHYAY

***Abstract*—This correspondence deals with the problem of testing the diagnosability of a system in presence of hybrid faults (that is, when some of the units of the system have failed intermittently and some have failed permanently). Presence of intermittent faults can lead to incomplete diagnosis and usually complicates the diagnosis problem in comparison to permanent fault situation. Alternative characterizations for hybrid fault diagnosability of a system to those originally presented in [3] are derived in this paper. It is shown that these conditions lead to the testing of the hybrid diagnosability of a system with fewer computations than that in [3].**

***Index Terms*—Hybrid faults, intermittent faults, permanent faults, system level diagnosis.**

## I. Introduction

The study of diagnosable systems under different models [1], [7]–[11] received considerable attention in recent years. The model introduced in [1] is possibly the most well-studied model in connection with diagnosability analysis under different measures. This model assumes a system to be composed of a number of units, each of which is tested by some other units of the system. Most studies on this model [1], [6]–[9], [12] assume that whenever any unit of the system is faulty, the fault is of permanent nature. In general, the fault situation is hybrid in the sense that some of the faulty units can fail intermittently while the others fail permanently. The problem of diagnosability of a system under intermittent or hybrid fault situation was studied in [2] and [3].

A system modeled as in [1] can be represented as follows. A system $S$ having $n$ units or components is represented by a directed graph $G$ with $n$ vertices. Each unit of the system is tested by some of the other units but never by itself. The feature of testing of each unit by other units is represented by directed edges of $G$. If the $i$th unit of the system tests the $j$th unit, then $G$ will have a directed edge from the $i$th vertex $v_i$ to the $j$th vertex $v_j$ and this edge will be represented by $(v_i, v_j)$. Throughout this correspondence, the vertices of $G$ and the corresponding units of $S$ will be used interchangeably. The result of testing of one unit by another is represented by labeling the edge representing the testing. An edge $(v_i, v_j)$ is labeled 0 if $v_i$ "finds" $v_j$ good and is labeled 1 if $v_i$ "finds" $v_j$ faulty. If $v_i$ is faulty, intermittently or permanently, its finding is unreliable and thus the labeling of $(v_i, v_j)$ does not convey any information about the status of $v_j$. If $v_i$ is fault free, then its finding about $v_j$ is reliable if $v_j$ is fault free or permanently faulty. If $v_j$ is intermittently faulty, the fault can escape detection by $v_i$ if $v_j$ was working in a fault free manner when $v_i$ tested it. Thus, if $v_j$ is intermittently faulty, the finding of $v_i$ about $v_j$ could be unreliable even when $v_i$ is fault free. Because the intermittently faulty units can escape detection by fault free units, the identification of faulty units from the test results will usually lead to incomplete identification. One way to reduce the incompleteness is possibly to repeat the testing several times with an expectation that the intermittently faulty units will show some evidence of fault during this repetition and will be ultimately detected. A system is said to be diagnosable if all the permanently faulty units and possibly some of the intermittently faulty units can be uniquely identified from the test results. In this correspondence, we will derive an alternative characterization to those originally presented in [3] for the diagnosability of a system assuming upper bounds on the number of faulty units and the number of intermittently faulty units. It will be shown that these characterizations lead to the testing of the diagnosability of a system with fewer computations than that in [3].

## II. Definitions and Notations

Let $G = (V, E)$ represent a system $S$ where $V$ is the set of vertices and $E$ is the set of directed edges of $G$, such that each vertex of $V$ represents an unit of $S$ and each edge represents the testing feature of one unit by another. Every element of $E$ is an ordered pair of vertices, that is, $E \subseteq V \times V$. Let $X(v_i)$ represent the set of vertices tested by $v_i$, i.e., $X(v_i) = \{v_j | (v_i, v_j) \in E\}$ and $X^{-1}(v_i) = \{v_j | (v_j, v_i) \in E\}$. For any $V_1 \subset V$, we define, $X(V_1) = \cup_{v_i \in V_1} X(v_i) - V_1$ and $X^{-1}(V_1) = \cup_{v_i \in V_1} X^{-1}(v_i) - V_1$.

Manuscript received July 9, 1983; revised July 2, 1984.

A. Sengupta and A. Sen are with the Department of Computer Science, University of South Carolina, Columbia, SC 29208.

S. Bandyopadhyay is with the Department of Computer Science, University of Windsor, Windsor, Ont., N9B 3P4, Canada.

IEEE Log Number 8406293.