Interchangeable Boolean Functions and their Effects on Redundancy in Logic Circuits

Debesh K. Das  
Dept. of Computer Science & Engg.  
Jadavpur University  
Calcutta - 700 032, India  
dgd@jadav.ernet.in

Susanta Chakraborty  
Dept. of Computer Science  
Kalyani University  
West Bengal, India  
susanta@klyuniv.ernet.in

Bhargab B. Bhattacharya  
Electronics Unit  
Indian Statistical Institute  
Calcutta - 700 035, India  
bhargab@isical.ernet.in

Abstract - A new concept of interchangeability of boolean functions under stuck-at faults in logic circuits is introduced in this paper. Two boolean functions F_1 and F_2 are said to be interchangeable if there exist two irredundant combinational networks N_1 and N_2 realizing F_1 and F_2 respectively, such that under some single/multiple stuck-at fault f_1(f_2) in N_1(N_2), the faulty network realizes F_2(F_1). It has been shown that an infinite family of such interchangeable pairs of functions exist, and they play an important role in determining several new types of redundancy in combinational and sequential circuits.

I. INTRODUCTION

In this paper, we introduce a new concept of interchangeability of boolean functions. Let N_1 and N_2 be two irredundant combinational networks realizing boolean functions F_1(x_1, x_2, ..., x_n) and F_2(x_1, x_2, ..., x_n) respectively. Let f_1(f_2) be a single/multiple stuck-at fault in N_1(N_2), such that the faulty network realizes F_2(F_1). Given a pair of functions (F_1, F_2), if there exist such networks and faults, then they are called an interchangeable or swapping pair of functions. We will show that such pairs play a major role in determining various types of redundancy in combinational as well as sequential circuits.

Combinational circuits realizing interchangeable functions may show some logical redundancy [9] under some stuck-at faults, even if the circuits are irredundant. Although these faults are detectable, removal of such logical redundancy leads to further optimization of the circuit. In the case of sequential circuits, existence of interchangeable logic functions on certain pair of internal lines may render some faults isomorph-redundant [5], and hence undetectable. We also show an example where interchangeable functions introduces a new kind of logical redundancy called p-isomorph redundancy, in a sequential circuit. Thus, knowledge of circuit behavior in the presence of interchangeable functions, might be useful for further logic optimization and avoidance of undetectable faults.

II. PRELIMINARIES

We will consider gate-level descriptions of logic circuits and classical stuck-at fault model.
ourselves to irredundant circuits only. It can be noted that
the number of possible irredundant realizations of an n-
variable function is also unknown though finite for n>2 [4].
Moreover, in a multi-output irredundant network, the sub-
circuit corresponding to a particular output may be irredu-
dant [3], for which we consider single-output circuits only.

III. INTERCHANGEABILITY IN TWO-LEVEL CIRCUITS

Let a boolean function F 1 be realized by a combinational
network N. If some stuck-at fault f (single/multiple) in
N, now changes F 1 to a function F 2 , we symbolize the event
as: F 1 =⇒ F 2 . F 2 is then said to be a faulty function
derivable from the fault-free function F 1.

Theorem 1: Let F 1 =⇒ F 2 and F 2 =⇒ F 3 hold. If F 2 is a
unique function, then F 1 =⇒ F 3.

Proof: Let F 1 =⇒ F 2 and F 2 =⇒ F 3 hold for some faults
f 1 and f 2 in their two-level AND-OR realizations N 1 and
N 2 respectively. Since, F 1 =⇒ F 2 , N 1 contains an irredu-
dant two-level subnetwork N′ (obtained by removing lines
and/or gates affected by faults) that realizes F 2 . A unique
function has only one irredundant two-level AND-OR
subnetwork N′. Therefore, the multiple fault {f 1 , f 2 } in N 1 will change F 2 to F 3.

Definition 4: Let F 1 =⇒ F 2 and F 2 =⇒ F 3 ,...,F k =⇒ F k+1
in irredundant networks N 1 ,N 2 ,...,N k realizing F 1 ,F 2 ,...,F k
respectively. Then a directed cycle of length k is said to exist among F 1 ,F 2 ,...,F k.
The presence of a directed cycle among the set of boolean
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1.
Such a structure is often used for parity checking, signature
analysis and self-checking. Now let under some faults, say
f 1 ,f 2 ,f 3 in N 1 ,N 2 ,N 3 respectively, one observes:

F 1 =⇒ F 2 =⇒ F 3 =⇒ F 1.

(i.e., a directed cycle is present among F 1 ,F 2 ,F 3).

Example 2: Consider the following three functions:

F 1 = x 1 x 2 + x 2 x 3 + x 3 + x 1 x 2 x 3
F 2 = x 1 x 2 + x 2 x 3 + x 3 + x 1 x 2 x 3
F 3 = x 1 x 2 + x 2 x 3 + x 3 + x 1 x 2 x 3

which are realized by two-level irredundant AND-OR
networks N 1 ,N 2 and N 3 (Figs. 2, 3, and 4) respectively. Now
denote by l d , the fault "line l stuck-at d \in (0,1)". One can
verify that: F 1 =⇒ F 2 =⇒ F 3 =⇒ F 1 . So, a directed
cycle of length 3 exists among F 1 ,F 2 ,F 3 . Also, as F 1 =⇒ F 3 ,
F 3 =⇒ F 1 , a directed 2-cycle is present between F 1 and F 3.

Theorem 2: The necessary conditions for the existence of
d symbol of that function. Thus, N 1 ,N 2 ,N 3 ,...,N k realizing F 1 ,F 2 ,...,F k respectively. Now
Consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
functions may have many far reaching consequences. For
example, consider a circuit structure as shown in Fig 1. Each of the
then they must be interchangeable.

**Example 4:** Let \( F_3 = x_1 x_2 + x_1 x_4 + x_2 x_1 + x_4 + x_2 + x_1 x_4 + x_1 \), then \( F_4 = x_1 x_2 + x_1 x_4 + x_1 x_4 + x_2 x_1 + x_4 + x_2 + x_1 x_4 + x_1 \) are interchangeable.

Corollary 3: There exists an infinite family of interchangeable functions.

IV. INTERCHANGEABILITY AND LOGICAL REDUNDANCIES

Two new kinds of redundancy called p- and n-redundancy were introduced in [9].

**Definition 6:** [9] A network \( N \) is **p-redundant** if the same function can be realized by introducing some stuck-at faults in \( N \), and permuting some input literals.

Remark: A p-redundant fault in an irredundant network is detectable. Thus, it is not a redundant fault in conventional sense. Elimination of these redundancies leads to further optimization in the circuit, and reduces complexity in test generation. It is some sort of logical redundancy.

**Example 5:** The network shown in Fig. 4 realizing \( F_4 = x_1 x_2 + x_1 x_4 + x_2 x_1 + x_4 + x_2 + x_1 x_4 + x_1 \), is p-redundant, because, the multiple fault \( \{ p, q \} \) in \( N_4 \) yields a faulty function \( F_4' = x_1 x_2 + x_1 x_4 + x_2 x_1 + x_4 + x_2 + x_1 x_4 + x_1 \). If we now interchange \( x_1 \) and \( x_3 \) in \( F_4 \), we get \( F_4 = F_3 \). Thus, \( F_3 \) and \( F_4 \) are interchangeable.

V. INTERCHANGEABILITY IN SEQUENTIAL MACHINES

A. Faults in sequential machines

We will consider sequential machines operating in free mode with unrestricted test strategy [6]. Let us consider a fault in a sequential machine \( M = \{ I, O, S, \delta, Z \} \), and let the faulty machine be denoted by \( M_f = \{ I, O, S_f, \delta_f, Z_f \} \).

A fault \( f \) is said to be **detectable** if, for every pair of initial states \( S \) and \( S' \) of the fault-free and faulty circuits respectively, there exists an input sequence \( X \), such that the response \( Z(X, S) \) of the fault-free circuit to \( X \), is different from the response \( Z_f(X, S') \) of the faulty circuit at some time unit on some output [6]. A fault is said to be **undetectable** if it is not detectable. A fault may change the state table of the machine, but remain undetectable. However, if initial operating condition of the machine is changed, this undetectable fault may be detected. Thus there may exist some undetectable faults which are **irredundant**. A subset of sequentially undetectable faults that are undetectable even under full scan, is called **combinatorially redundant**.

In this paper, we focus on a special kind of sequentially undetectable faults known as **isomorph fault**.

**Definition 7** [5]: A fault \( f \) in a reduced, completely specified, sequential machine, is an **isomorph fault** if the state table of the faulty machine is identical to that of the fault-free machine under some permutation of the states.

B. Interchangeability and isomorph-redundancy

Isomorph-redundancies in sequential circuits was introduced in [7]. Examples of these redundancies and their classification were first given in [8].

Interchangeable functions play an important role in producing isomorph-redundancies.

**Example 6:** Consider two function \( F_1 \) and \( F_2 \) of Example 4, as realized by two networks \( N_1 \) and \( N_2 \) of Figs. 4 and 5 respectively. They are P-equivalent, because by permuting \( x_1 \) and \( x_3 \) in \( F_1 \), we get \( F_1 \). Now the multiple fault \( \{ p, q \} \) in \( N_1 \) yields a faulty function \( F_1' = x_1 x_2 + x_1 x_4 + x_2 x_1 + x_4 + x_2 + x_1 x_4 + x_1 \). Similarly, the corresponding multiple fault \( \{ p, q \} \) in \( N_2 \) creates \( x_1 x_2 x_4 + x_2 x_1 x_4 + x_1 x_2 x_4 + \), which is again identical to \( F_1 \). Thus \( F_1 \) and \( F_2 \) are interchangeable.

Lemma 2: If \( F(x_1, x_2, \ldots, x_n) \) and \( F(x_1, x_2, \ldots, x_n) \) are interchangeable with respect to two irredundant realizations \( N_1 \) and \( N_2 \), then there exists a two-level irredundant realization of \((pF_i + qF_j)\) that is p-redundant, where \( p \) and \( q \) are variables distinct from \( x_1, x_2, \ldots, x_n \).

**Proof:** As \( p \) and \( q \) are distinct, obviously there exists a multiple fault \((m, n)\) and an irredundant realization of \((pF_i + qF_j)\), with \( F_i \) realized by \( N_1 \) and \( N_2 \), s.t. \((pF_i + qF_j) \rightarrow (qF_i + pF_j)\). Obviously, \( N \) is p-redundant, as permuting \( p \) and \( q \) in faulty function \((qF_i + pF_j)\), we get the original function \((pF_i + qF_j)\).

**Lemma 3:** There exists an infinite class of p-redundant circuits.

**Proof:** Follows from Lemma 2 and Corollary 3.
respectively. They are used to design a combinatorially irredundant sequential circuit as shown in Fig. 6. It can be shown that the machine \( M \) (state table shown in Table 1) corresponding to this circuit is reduced. Now, for the multiple fault shown in Fig. 6, that changes \( Y_1 \) to \( Y_2 \), and \( Y_2 \) to \( Y_1 \), let the faulty machine be \( M' \) (state table shown in Table 2). It is now easy to observe that \( M \) and \( M' \) are isomorphic to each other, with respect to swapping of the states \( S_1 \) (01) and \( S_2 \) (10).

**Remark:** Consider two combinational circuits \( N_i \) and \( N_j \) realizing two interchangeable functions \( F_i \) and \( F_j \) respectively, where \( F_i \) and \( F_j \) are \( P \)-equivalent. Let \( f_1 \) (\( f_2 \)) denote the fault in \( N_i \) (\( N_j \)), s.t. the faulty function becomes \( F_j \) (\( F_i \)) respectively. Then the machine of Fig. 7 is isomorph-

<table>
<thead>
<tr>
<th>Table 1: State table of the fault-free machine of Fig. 6 using ( N_3 ) and ( N_4 ) realizing ( F_3 ) and ( F_4 ) respectively</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal equivalents of inputs: ( x_4 x_3 x_2 x_1 )</strong></td>
</tr>
<tr>
<td>( Y_{Y_2} )</td>
</tr>
<tr>
<td>( S_0(00) )</td>
</tr>
<tr>
<td>( S_1(01) )</td>
</tr>
<tr>
<td>( S_2(10) )</td>
</tr>
<tr>
<td>( S_3(11) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: State table of the faulty machine of Fig. 6 in which ( N_3 ) and ( N_4 ) realize ( F_4 ) and ( F_3 ) respectively</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal equivalents of inputs: ( x_4 x_3 x_2 x_1 )</strong></td>
</tr>
<tr>
<td>( Y_{Y_2} )</td>
</tr>
<tr>
<td>( S_0(00) )</td>
</tr>
<tr>
<td>( S_1(01) )</td>
</tr>
<tr>
<td>( S_2(10) )</td>
</tr>
<tr>
<td>( S_3(11) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3: State table of the fault-free machine of Fig. 10 where ( N_5 ) (( N_6 )) realizes ( F_1 ) (( F_5 )) in the fault-free machine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal equivalents of inputs: ( x_4 x_3 x_2 x_1 )</strong></td>
</tr>
<tr>
<td>( Y_{Y_2} )</td>
</tr>
<tr>
<td>( S_0(00) )</td>
</tr>
<tr>
<td>( S_1(01) )</td>
</tr>
<tr>
<td>( S_2(10) )</td>
</tr>
<tr>
<td>( S_3(11) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4: State table of the faulty machine of Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal equivalents of inputs: ( x_4 x_3 x_2 x_1 )</strong></td>
</tr>
<tr>
<td>( Y_{Y_2} )</td>
</tr>
<tr>
<td>( S_0(00) )</td>
</tr>
<tr>
<td>( S_1(01) )</td>
</tr>
<tr>
<td>( S_2(10) )</td>
</tr>
<tr>
<td>( S_3(11) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5: State table of the faulty machine of Table 3 with swapping of ( x_1 ), ( x_2 ), and ( x_3 ), ( x_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal equivalents of inputs: ( x_4 x_3 x_2 x_1 )</strong></td>
</tr>
<tr>
<td>( Y_{Y_2} )</td>
</tr>
<tr>
<td>( S_0(00) )</td>
</tr>
<tr>
<td>( S_1(01) )</td>
</tr>
<tr>
<td>( S_2(10) )</td>
</tr>
<tr>
<td>( S_3(11) )</td>
</tr>
</tbody>
</table>
redundant under the multiple fault \((f_1, f_2)\). Removal of this redundancy leads to a reduced network.

**Theorem 5:** There exists an infinite family of isomorph-redundant circuits.

**Proof:** Follows from the fact that the machine of Fig. 7 is isomorph-redundant, and there exists an infinite class of p-redundant circuits (Lemma 3). 

**C. Interchangeability and p-isomorph redundancy**

In the case of combinational circuits, we call a circuit to be p-redundant, if the faulty-free and faulty functions are P-equivalent under a fault. For sequential machines, these p-redundancies are also valid, where removal of the effect of a fault, and a subsequent interchange of some input literals would realize the same machine with less hardware.

**Definition 8:** A sequential circuit \(C\) is p-isomorph redundant if the presence of some stuck-at faults in \(C\), followed by permutation of some input literals in the faulty circuit, produces a machine isomorphic to the original one.

Interchangeable functions also play a role in producing p-isomorph redundancy in a sequential circuit.

**Example 7:** Consider two functions \(F_1 = x_1 x_3 + x_2 x_4 + x_3 x_5\) and \(F_2 = x_1 x_3 + x_2 x_4 + x_3 x_5\), as realized by the irredundant networks \(N_1\) and \(N_2\) (Figs. 8 and 9). They are used to form a combinatorially irredundant sequential network of Fig. 10, whose state table is shown in Table 3. Notice that the machine is reduced. For the multiple fault \(\{l_1, l_2\}\), the corresponding faulty machine is shown in the state-table of Table 4. Now, if we interchange \(x_1, x_3, x_2, x_4\) in the faulty machine, we get the machine of Table 5. Machines of Tables 3 and 5 are isomorph to each other, with respect to swapping of the states \(S_1\) (01) and \(S_2\) (10).

Notice that the fault \(l_1^1(l_2^1)\) in the subnetwork \(N_4(N_5)\) of Fig. 10, yields a faulty function \(F'_1(F'_2)\). Furthermore, \(F'_1(F'_2)\) is P-equivalent to \(F_2(F_1)\).

Consider the network of Fig. 11, where, \(F_1(x_1, x_2, \ldots, x_n)\) and \(F_2(x_1, x_2, \ldots, x_n)\) are interchangeable functions, and the literals \(p\) and \(q\) are distinct from \((x_1, x_2, \ldots, x_n)\). It is easy to show that this network is p-isomorph redundant.

**Corollary 4:** There exists an infinite family of p-isomorph redundant circuits.

**Proof:** Follows from Corollary 3. 

**Remark:** Although an isomorph-redundant fault is always undetectable, a p-isomorph redundant fault is detectable, and hence not redundant in conventional sense. However, elimination of affected lines and gates, and subsequent permutation of primary inputs restores an isomorphic and hence an equivalent machine.

**VI. CONCLUSION**

This paper introduces a new concept of interchangeable functions in combinational circuits. It has been shown that these functions play an important role in producing new redundancies such as p-redundancy in combinational circuits, and isomorph- as well as p-isomorph redundancy in sequential circuits. Though p-redundant and p-isomorph redundant stuck-at faults are otherwise detectable, their removal decreases the cost of the circuit and therefore, reduces the complexity in test generation. Isomorph-redundant faults are however undetectable, and thus create more problems than the above two logically redundant faults. No efficient procedure for identifying and removing isomorph-redundant faults is yet known. We have shown that an infinite class of such redundant circuits can be built using interchangeable functions. Further investigation in this direction is needed to analyze and characterize such redundancies in logic circuits.

**REFERENCES**

Fig. 1: A circuit used for parity checking

Fig. 2: Circuit $N_1$ realizing $F_1$

Fig. 3: Circuit $N_2$ realizing $F_2$

Fig. 4: Circuit $N_3$ realizing $F_3$

Fig. 5: Circuit $N_4$ realizing $F_4$

Fig. 6: A combinationally irredundant sequential circuit with an isomorph-redundant fault

Fig. 7: A circuit structure having isomorph-redundancy $(F_i$ and $F_j$ are $P$-equivalent and interchangeable)

Fig. 8: Circuit $N_5$ realizing $F_1$

Fig. 9: Circuit $N_6$ realizing $F_5$

Fig. 10: A $p$-isomorph-redundant circuit

Fig. 11: A circuit having $p$-isomorph-redundancy $(F_i$ and $F_j$ are interchangeable functions)