JOURNAL OF ELECTRONIC TESTING: Theory and Applications 8, 71–86 (1996)
© 1996 Kluwer Academic Publishers. Manufactured in The Netherlands.

Balance Testing and Balance-Testable Design of Logic Circuits*

KRISHNENDU CHAKRABARTY
Department of Electrical, Computer and Systems Engineering, Boston University, 44 Cummington Street, Boston, MA 02215
kchakrab@hilsa.bu.edu

JOHN P. HAYES
Advanced Computer Architecture Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122

Received November 4, 1994; Revised September 7, 1995

Editor: Y. Zorian

Abstract. We propose a low-cost method for testing logic circuits, termed balance testing, which is particularly suited to built-in self testing. Conceptually related to ones counting and syndrome testing, it detects faults by checking the difference between the number of ones and the number of zeros in the test response sequence. A key advantage of balance testing is that the testability of various fault types can be easily analyzed. We present a novel analysis technique which leads to necessary and sufficient conditions for the balance testability of the standard single stuck-line (SSL) faults. This analysis can be easily extended to multiple stuck-line and bridging faults. Balance testing also forms the basis for design for balance testability (DFBT), a systematic DFT technique that achieves full coverage of SSL faults. It places the unit under test in a low-cost framework circuit that guarantees complete balance testability. Unlike most existing DFT techniques, DFBT requires only one additional control input and no redesign of the underlying circuit is necessary. We present experimental results on applying balance testing to the ISCAS 85 benchmark circuits, which show that very high fault coverage is obtained for large circuits even with reduced deterministic test sets. This coverage can always be made 100% either by adding tests or applying DFBT.

Keywords: built-in self testing, design for testability, fault coverage, fault detection, testing methods

1. Introduction

Built-in self testing (BIST) techniques aim to reduce testing cost and improve test quality by means of on-chip test generation and response verification circuitry [1]. Response observation in BIST is usually done by saving the test outcome in a compressed form called a signature. A circuit is tested by comparing the observed signature with the correct fault-free signature. The process of reducing the complete test response to a signature is referred to asresponse compression. Some well-known compression methods are ones counting [2], parity checking [3], transition counting [2], syndrome testing [4], and signature analysis [5].

A problem with most compression techniques is that it is difficult or impractical to determine precisely the

*This research was supported by the National Science Foundation under Grant No. MIP-9200526. Parts of this paper were published in preliminary form in *Proc. 23rd Symp. Fault-Tolerant Computing*, Toulouse, June 1993, and in *Proc. 31st Design Automation Conf*, San Diego, June 1994.

fraction of faults of interest that are detected or covered. The calculation of fault coverage in compression testing is especially difficult since the coverage depends not only on the test set, but also on the compression technique employed. The relationship between the compression function and fault detection is complex and poorly understood. Compression techniques are analyzed using error models which, although useful for comparing the different methods, provide little general information about the fault coverage. Furthermore, every circuit under test has a different fault-free signature that has to be computed and stored.

We propose a new testing technique termed *balance testing*, which provides high fault coverage and is easy to implement; it is therefore particularly attractive for BIST. The motivation for balance testing arises from balanced functions, a class of Boolean functions that we have identified and analyzed [6]. Faults are detected by checking the test response for the balance property by means of a counter at the output of the circuit under test.

Balance testing offers several advantages over existing compression techniques. It is relatively easy to specify necessary and sufficient conditions for the detectability of single stuck-line (SSL) faults. These conditions, which can be easily extended to multiple stuck-line (MSL) and bridging faults, are useful because they enable the designer to identify the faults that remain undetected by balance testing. This is a significant advance over previous BIST techniques, for which little can be said about the conditions under which a fault is detectable. Our analysis also obviates the need for error models that are unrelated to physical faults and are usually difficult to validate. To deal with balance-untestable faults, we develop a systematic design for testability (DFT) method that guarantees full coverage of SSL faults. Finally, the fault-free signature for balance testing is always the all-0 pattern, so there is no need to compute and store different fault-free signatures.

The organization of the paper is as follows. Section 2 describes balanced functions and introduces balance testing. In Section 3, we derive necessary and sufficient conditions for the balance testability of SSL, MSL and bridging faults. Section 4 describes the new DFT technique to eliminate balance-untestable faults. Finally, in Section 5, we discuss some applications of balance testing to large, multiple-output circuits, and present experimental results for the ISCAS 85 benchmarks.

2. Balanced Functions and Testing

Boolean functions that are true for exactly half their input combinations and false for the other half are called *balanced* [6]. Consider an exclusive-or gate with n inputs that realizes the odd parity function $f(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$. The output of f is 1 when the number of 1's applied to the gate is odd; this clearly occurs in half the possible cases, implying that f is balanced. A surprisingly large number of other common functions are also balanced, including the sum and carry functions of an adder, the 2^n-to-1 multiplexer function, and the next-state function of D and *JK* flip-flops. Let $f(x_1, x_2, \ldots, x_n) = f(X)$ be a Boolean function in n variables. We call X a *true vector* of f if $f(X) = 1$, and a *false vector* if $f(X) = 0$. Let $|f|^1$ denote the number of true vectors in f, and let $|f|^0$ denote the number of false vectors in f. An n-variable Boolean function $f(x_1, x_2, \ldots, x_n)$ is balanced if and only if it has an equal number of true and false vectors, i.e., $|f|^1 = |f|^0 = 2^{n-1}$. The functional properties of balanced functions are studied further in [6].

Special classes of Boolean functions and circuits realizing these functions have been the focus of much research in the past. Their distinctive properties often simplify the problem of testing them. Fanout-free combinational circuits are well-known examples of special circuits that are easily testable [7, 8, 3]. Balanced circuits also have very desirable testing properties, but are more widely applicable, as we demonstrate in this paper.

In 1958, Kautz [9] introduced the term "neutral" for what we have independently called balanced functions. The use of neutral functions was limited to counting the number of symmetry classes of self-complementary functions [10] and no practical applications were developed. Balanced functions also appear in an implicit and unnamed form in some recent testing research. Chatterjee and Abraham [11] present a unified testing theory for arithmetic logic arrays, in which balance plays a key role. For example, Lemma 1 of [11] states that for a tree of identical, single-output cells to be minimally C-testable, it is necessary that the number of ones observed on the output of a cell when all input combinations are applied to its inputs must be identical to the number of zeros. Kundu [12] shows that balance is necessary to completely characterize a generalized form of functional completeness.

We next describe *balance testing*, which detects faults by checking test response sequences for the

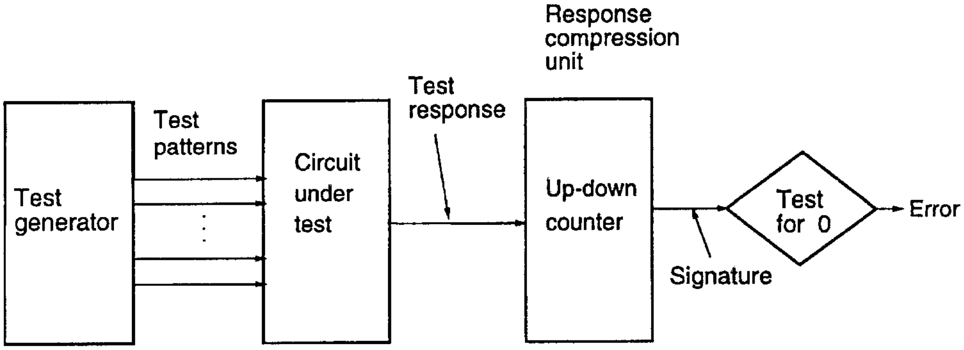

Figure 1. A typical balance testing scheme.

balance property. This can be easily accomplished for a circuit that implements a balanced function. For example, the test response R can be directed to an up-down counter as shown in Fig. 1. The counter is incremented whenever R is 1, and decremented whenever R is 0. A fault is detected if and only if the counter's final state (the signature) is nonzero. To apply balance testing to an unbalanced circuit, we can replace the up-down counter of Fig. 1 by a down counter preset to $|f|^1$, where $|f|^1$ is the number of minterms of the function f realized by the circuit. The counter is decremented whenever the test response is 1. Balance testing is directly applicable to combinational circuits of moderate size, and can be extended to large circuits via logic partitioning, scan design, or the like [1]. The application of balance testing to large, multiple-output circuits is discussed in Section 5.

A possible approach to balance testing is to use exhaustive tests, i.e., to apply all possible input patterns to the circuit under test. We refer to this approach as *exhaustive* balance testing. Some other BIST techniques such as syndrome testing [4] also employ exhaustive testing. The advantages of exhaustive testing include ease of test generation, the fact that all detectable combinational faults are sensitized, and amenability to formal analysis [13]. For circuits with a large number of primary inputs (20 or more), pseudoexhaustive testing may be employed using partitioning techniques to reduce its complexity. Pseudoexhaustive testing has received considerable attention recently, and a number of efficient logic partitioning tools have been developed for it [14, 15, 16]. For multiple-output circuits, either the testing process can be repeated for every observable output, or a suitable space compaction circuit can be used [17, 18].

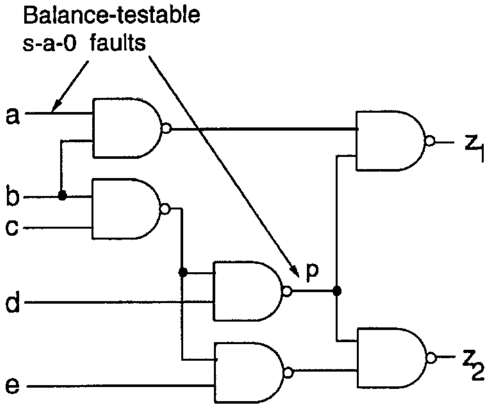

Figure 2. The ISCAS c17 benchmark circuit with two balance-testable SSL faults marked.

However, exhaustive balance testing is not necessary to obtain good fault coverage; we can also use pseudorandom or "reduced" deterministic (nonrandom) test sets, as we will demonstrate. Pseudorandom patterns can be generated using linear feedback shift-registers [1], while reduced test sets can be obtained using an ATPG program and applied using either a ROM and counter or a nonlinear feedback shift-register [19]. For nonexhaustive balance testing, the down counter is preset to the number of 1s in the fault-free test response and, as before, it is decremented whenever the test response is 1.

Consider the small 5-input ISCAS benchmark circuit c17 [20] shown in Fig. 2. There are two outputs z_1 and z_2, and the number of true vectors for each output is 18. Therefore, to generate a zero fault-free signature for z_1 or z_2 with exhaustive testing, the down counter must be preset to 18. Consider the fault a stuck-at-0 (s-a-0). The number of true vectors for the output z_1

Table 1. SSL faults detected by exhaustive balance testing for some useful logic circuits.

Circuit	Circuit description	Number of SSL faults	Number balance-testable	Percent balance-testable
7442	Decoder	168	168	100
7485	Comparator	228	228	100
74147	Priority encoder	190	190	100
74148	Priority encoder	226	226	100
74150	Multiplexer	412	404	98
74181	ALU	384	376	98
74182	Carry-lookahead generator	184	184	100
74280	Parity generator	240	198	83
74283	Carry-lookahead adder	250	142	57

in the faulty case is 20, which gives a balance signature of $18 - 20 \pmod{2^5} = 30$; therefore this fault is balance-testable. (In fact, all SSL faults in this circuit are balance-testable, as can readily be verified by hand or computer simulation.) In Table 1, we list the number of balance-testable SSL faults for some common MSI logic circuits in the 74X series [21]. The 74150 multiplexer circuit, for example, contains eight balance-redundant faults; these are the SSL faults on the four select inputs.

The fault-free signature in balance testing is always the all-0 pattern. This zero signature is easy to detect, and there is no need to store a precomputed fault-free reference signature. The test outcome is independent of the order of application of test patterns, therefore the test controller is also easy to implement. A previous zero-signature method for BIST uses linear feedback-shift registers (LFSRs) to compress the test response [22]. Balance testing is related to this signature analysis technique in that the preset value of the down counter corresponds to the initial state (seed) of the LFSR. However, balance testing differs from [22] in that it uses ordinary counters instead of LFSRs, and while the preset value of the counter is obtained via fault-free simulation, the LFSR's seed is derived from its autonomous behavior. Balance testing also leads to a direct analysis of fault coverage using functional properties of the circuit under test. (The fault detection capabilites of the zero-signature scheme of [22] have not been analyzed.) Moreover, it is possible to achieve full fault coverage in balance testing using the DFT technique described in Section 4.

It is also useful to compare balance testing to syndrome testing [4]. Both schemes employ ones counting and use a counter at the circuit output. Balance testing is an application of balanced functions; thus, the analysis of balance testability can be based on a technique for analyzing functional composition that was developed in [6]. In [4], only SSL faults are addressed with respect to their syndrome testability, whereas in this paper, we characterize the balance testability of not only SSL faults, but also of MSL and bridging faults. Since syndrome testing computes a signature of the form $|f|^1/2^n$ for an n-variable function f, it can be regarded as a variant of exhaustive balance testing with a normalized signature and a preset value of zero. Hence, our results on balance testing are also applicable to syndrome testing.

A problem associated with balance testing (and with most other response compression schemes) is that not all faults disturb the tested property of the circuit—in this case, balance—and so remain undetected. Table 1 gives the number of detectable SSL faults for some 74X-series circuits. We next characterize balance-untestable or "redundant" faults.

3. Balance Testability

In this section, we derive necessary and sufficient conditions for the balance testability of SSL faults in single-output circuits. Unless otherwise stated, we assume that exhaustive balance testing is employed. We also assume throughout that all faults considered are detectable (non-redundant) in the usual sense.

Let $F(X)$ be the function realized by the circuit under test (CUT) and let $|X| = N$. Let $p = f(X_1)$ be any line in the circuit. We can express $F(X)$ as $g(f(X_1), X_2)$, where $X_1 \cup X_2 = X$ (see Fig. 3). Let $|F|^1 = 2^{N-1} + \delta$, where δ is the *balance offset* of F; we also refer to δ as the offset of the CUT. Clearly, $0 \le |\delta| \le 2^{N-1}$, and for balanced circuits, $\delta = 0$. For example, the function $f(x_1, x_2, x_3) = x_1x_2x_3$ has $|f|^1 = 1$, therefore its balance offset is $\delta = 1 - 4 = -3$.

A fault p s-a-d is *balance-redundant* if it changes the fault-free function $F(X)$ to a faulty function $F^\star(X)$ with the same balance offset. Balance-redundant faults are not detected by balance testing. Faults that make $F^\star(X)$ unbalanced are termed *balance-testable.*

The following theorem provides a necessary and sufficient condition for the balance testability of SSL

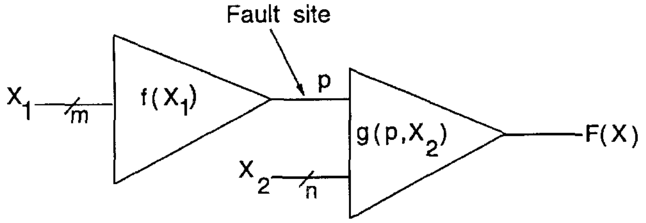

Figure 3. Notation used to analyze balance redundancy of SSL faults.

faults. Let p be 1 in k_p^g true vectors of $g(p, X_2)$ and 0 in $k_{p'}^g$ true vectors of $g(p, X_2)$. In other words, $k_p^g = |g(1, X_2)|^1$ and $k_{p'}^g = |g(0, X_2)|^1$. For example, if $g(p, x_3) = p \oplus x_3$, then $k_p^g = k_{p'}^g = 1$. We can combine k_p^g and $k_{p'}^g$ as follows: $k_{p^d}^g = k_p^g$ if $d = 1$, and $k_{p^d}^g = k_{p'}^g$ if $d = 0$.

Theorem 1. *Let a circuit with balance offset δ realize $F(X_1 \cup X_2) = g(f(X_1), X_2)$ with $p = f(X_1)$. The fault p s-a-d is balance-redundant if and only if $k_{p^d}^g = 2^{n-1} + \delta/2^{m-w}$, where (i) $|X_1| = m$, $|X_2| = n$, $|X_1 \cap X_2| = w$, and (ii) $k_{p^d}^g$ is the number of true vectors of $g(d, X_2)$.*

Proof: Let $X_1 \cap X_2 = \{x_1, x_2, \ldots, x_w\}$. We define the parameter $k^g_{p^d, x_1^{d_1}, \ldots, x_w^{d_w}}$, where $d_i, d \in \{0, 1\}$, along the same lines as $k_{p^d}^g$. We can show that

$$\begin{aligned}|F|^1 &= \sum_{d_1,\ldots,d_w} k^g_{p, x_1^{d_1}, \ldots, x_w^{d_w}} k^f_{x_1^{d_1}, \ldots, x_w^{d_w}} \\ &\quad + \sum_{d_1,\ldots,d_w} k^g_{p', x_1^{d_1}, \ldots, x_w^{d_w}} k^{f'}_{x_1^{d_1}, \ldots, x_l^{d_w}} \\ &= 2^{m+n-w-1} + \delta\end{aligned}$$

Consider the fault p s-a-0. Let $f^\star$ be the faulty function corresponding to f in the presence of this fault. Then $|f^\star|^1 = 0$ and $|f^\star|^0 = 2^m$; therefore, $k^f_{x_1^{d_1}, \ldots, x_w^{d_w}} = 0$ and $k^{f'}_{x_1^{d_1}, \ldots, x_w^{d_w}} = 2^{m-w}$. This gives us $|F^\star|^1 = 2^{m-w} \sum_{d_i} k^g_{p', x_1^{d_1}, \ldots, x_w^{d_w}}$. Hence for the fault to be balance-redundant, $2^{m-w} \sum_{d_i} k^g_{p', x_1^{d_1}, \ldots, x_w^{d_w}} = 2^{m+n-w-1} + \delta$, which implies that $\sum_{d_i} k^g_{p', x_1^{d_1}, \ldots, x_w^{d_w}} = 2^{n-1} + \delta/2^{m-w}$, i.e., $k_{p'}^g = 2^{n-1} + \delta/2^{m-w}$. The proof for p s-a-1 is similar. ■

Observe that the balance redundancy of p s-a-d depends only on the function $g(p, X_2)$ and is independent of its implementation. It is also independent of the function $f(X_1)$. In addition, if F is balanced, then $\delta = 0$, and therefore balance redundancy is independent of the overlap between X_1 and X_2. An intuitive explanation for this is that balance is a functional property, and since all combinations of X_1 are applied to the circuit under test an equal number of times ($|X_2 - X_1 \cap X_2|$), the function $f(X_1)$ does not affect the balance testability of p s-a-d. However, generally not all combinations of $\{p, X_2\}$ appear an equal number of times on the inputs to $g(p, X_2)$, therefore it is not intuitively clear that the balance testability of p s-a-d is independent of the implementation of $g(p, X_2)$.

For example, in the circuit of Fig. 4(a), $g(p, c) = p \oplus b \oplus c$, $n = 2$, and p depends on b. Since $k_p^g = 2 = 2^{n-1}$, the faults p s-a-0 and s-a-1 are balance-redundant. In the 2-to-1 multiplexer of Fig. 4(b), it is easy to see that the faults p s-a-0 and s-a-1 are balance-testable. Figure 4(c) shows a circuit where the s-a-0 fault on the line p is balance-testable, while the s-a-1 fault is balance-redundant.

Theorem 1's condition for an SSL fault to be balance-redundant imposes severe constraints on the circuit under test. As a result, most SSL faults can be expected to be balance-testable, an observation supported by the experimental data of Table 1. Theorem 1 also implies the following useful necessary conditions for an SSL fault in the CUT to be balance-redundant: (i) $|\delta| \geq 2^{m-w}$, and (ii) If $m \neq w$, i.e., X_1 and X_2 are at least partially disjoint, then $|\delta|$ must be a power of 2; this leads to the following sufficient condition for balance testability.

Corollary 1. *Let a circuit with balance offset δ realize $F(X) = g(p(X_1), X_2)$. If δ is odd and $X - X_2 \neq \phi$, then the faults p s-a-1 and p s-a-0 are balance-testable.*

Functions realized by fanout-free circuits have an odd number of true vectors [3], implying that $|\delta|$ is always odd. Therefore, since $w = 0$, all SSL faults

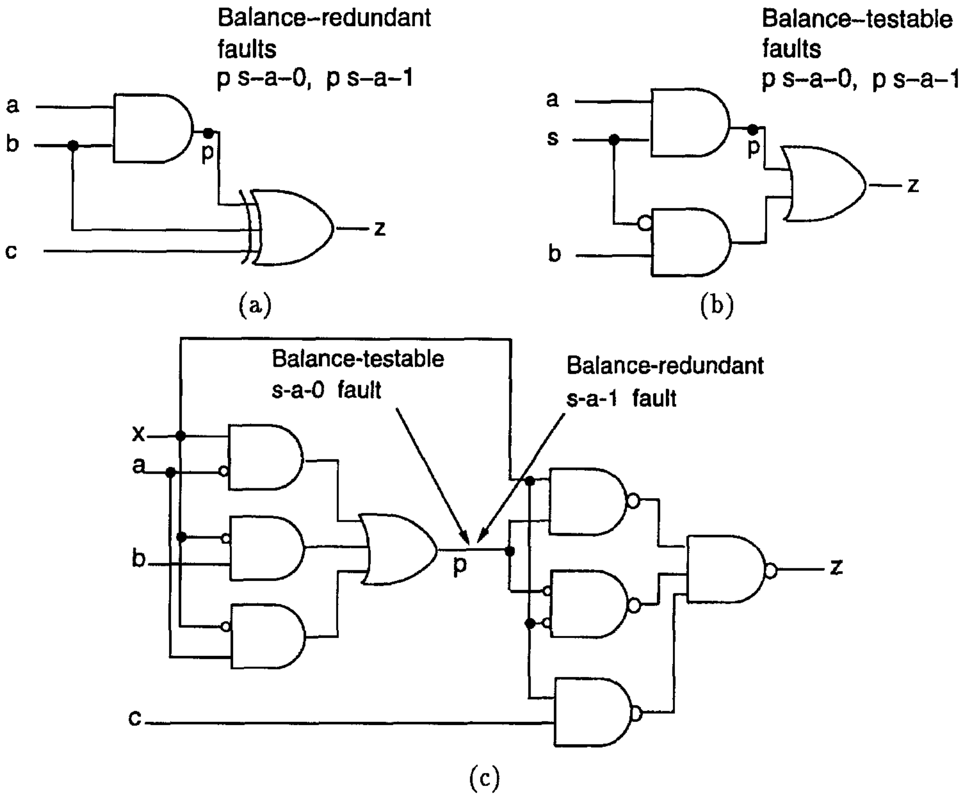

Figure 4. Balance redundancy of SSL faults: (a) balance-redundant faults; (b) balance-testable faults; (c) a line with both balance-redundant and balance-testable faults.

in fanout-free circuits are balance-testable. Savir (Lemma 6 in [4]) proves that all SSL faults of a fanout-free circuit are syndrome-testable. Corollary 1 demonstrates that the fanout-free property, though sufficient, is not necessary for either syndrome or balance testability.

Corollary 2. *Let a circuit with balance offset δ realize $g(p(X_1), X_2)$. If $|X_1| > \lceil \log_2 |\delta| \rceil + |X_1 \cap X_2|$, the faults p s-a-0 and p s-a-1 are balance-testable.*

Returning to the c17 benchmark circuit in Fig. 2, there are two outputs, each with $|F|^1 = 18$. This implies that the offset δ is 2 and $\log_2 |\delta| = 1$. For the fault p s-a-0 and output z_1, $X_1 = \{b, c, d\}$, and $X_2 = \{a, b\}$. Therefore $|X_1| = 3 > \log_2 |\delta| + |X_1 \cap X_2| = 2$, which implies that the fault is balance-testable.

For SSL faults on a primary input of the circuit under test, we have the following interesting corollary to Theorem 1.

Corollary 3. *Let x be a primary input of an irredundant circuit under test realizing the function F with balance offset δ. If x does not fan out and δ is odd, then the s-a-0 and s-a-1 faults on x are balance-testable. If x fans out, the s-a-0 and s-a-1 faults on x are balance-testable.*

We next consider why an up-down counter is not suitable for testing unbalanced circuits. If we use an up-down counter for explicit balance testing, the preset value should be $(2^N - 2\delta) \pmod{2^N}$. The justification for this preset value is as follows: If $|F|^1 = 2^{N-1} + \delta$, then $|F|^0 = 2^{N-1} - \delta$. Therefore, for the fault-free circuit, the up-down counter would register the value $((2^N - 2\delta) \pmod{2^N}) + |F|^1 - |F|^0) \pmod{2^N}) = 0$. As before, a fault is balance-testable if it makes the counter value different from zero. We now show that an up-down counter produces additional balance-redundant faults, and therefore a down counter should be used for response compression.

Lemma 1. *Let $F(x_1, x_2, \ldots, x_N)$ be the function realized by the CUT such that $|F|^1 = 2^{N-1} + \delta$. A fault that transforms F to $F^\star$, with $|F^\star| = 2^{N-1} + \delta + \sigma$, is not detected using an up-down counter whenever $\sigma = 0$ or $\sigma = \pm 2^{N-1}$.*

Proof: Let C be the value registered by the up-down counter for the faulty circuit. Now, $C = (2^N - 2\delta + 2^{N-1} + \delta + \sigma - 2^{N-1} + \delta + \sigma) \pmod{2^N} = 2\sigma \pmod{2^N}$. Thus, the fault is not detected if and only if $2\sigma = 0 \pmod{2^N}$, and since $|\sigma| \leq 2^{N-1}$, the lemma follows. ∎

If we use a down counter instead of an up-down counter, a fault is masked only when σ is 0. The following theorem, derived from Lemma 1, demonstrates that balance redundancy is more likely when an up-down down counter is used.

Theorem 2. *Let $F(X) = g(f(X_1), X_2)$ be the function realized by the CUT. Let $p = f(X_1)$, $|X_1| = m$, $|X_2| = n$, and $|X_1 \cap X_2| = w$. Let δ be the offset of $|F|^1$ from the balance point and let $k^g_{p^d}$ be the number of true vectors of $g(d, X_2)$. With an up-down counter, the fault p s-a-d is balance-redundant if and only if (a) for $\delta > 0$, $k^g_{p^d}$ is either $2^{n-1} + \frac{\delta}{2^{m-w}}$ or $\frac{\delta}{2^{m-w}}$, (b) for $\delta < 0$, $k^g_{p^d}$ is either $2^{n-1} + \frac{\delta}{2^{m-w}}$ or $2^n - \frac{\delta}{2^{m-w}}$.*

For example, suppose the CUT realizes a 4-input function $F(N = 4)$ with 12 true vectors ($|F|^1 = 12$, $\delta = 4$). Suppose a fault in the CUT produces the faulty function $F^\star$ with $|F^\star|^1 = 4$, i.e. $\sigma = -8$. Then the counter reading (signature) is $8+4-12 = 0$ which means that the error is not detected, and therefore the fault is balance-redundant.

So far, we have assumed that a modulo 2^N response compression counter is used for exhaustive balance testing of a circuit with N primary inputs. However, in order to reduce hardware overhead, a smaller, modulo q ($q < 2^N$) down counter can be used. The following theorem provides a necessary condition for balance testability in that case.

Theorem 3. *Let a circuit with balance offset δ realize $F(X_1 \cup X_2) = g(f(X_1), X_2)$ with $p = f(X_1)$, and suppose a modulo q response compression counter is used for balance testing. The fault p s-a-d is balance-redundant if and only if $k^g_{p^d} = 2^{n-1} + \frac{\delta+(v-c)q}{2^{m-w}}$, where (i) $|X_1| = m$, $|X_2| = n$, $|X_1 \cap X_2| = w$, (ii) $k^g_{p^d}$ is the number of true vectors of $g(d, X_2)$, and (iii) $c = \lfloor |F|^1/q \rfloor$ and v is any nonnegative integer.*

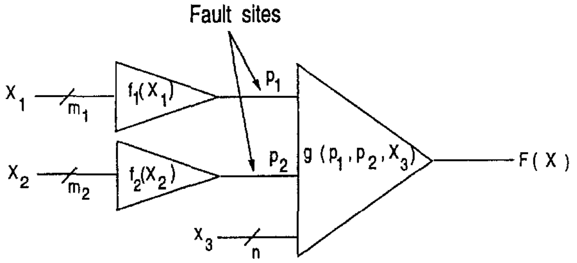

Figure 5. Notation used in characterizing balance redundancy of double stuck-line faults.

Proof: Following the notation of Theorem 1, the counter's preset value is $x = 2^{m+n-w-1} + \delta - cq$. Now, the fault p s-a-d is balance-redundant if and only if $2^{m-w} k^g_{p^d} = vq + x$, where v is any nonnegative integer. This implies that $k^g_{p^d} = 2^{n-1} + \frac{\delta+(v-c)q}{2^{m-w}}$. ∎

Theorem 3 is a generalization of Theorem 1 in the sense that the latter can be derived from it by setting $v = 0$ and $q = 2^N$, which in turn imply that $c = 0$. While it is intuitively obvious that fault masking, and therefore balance redundancy, is more likely with a smaller counter, Theorem 3 clearly specifies the values of $k^g_{p^d}$ that cause fault masking. If a modulo 2^N counter is used, fault masking occurs only when $|F^\star|^1 = |F|^1$. On the other hand, when a modulo q counter is used, the number of values of $|F^\star|^1$ for which masking occurs is $\lfloor \frac{2^N - (|F|^1 - cq)}{q} \rfloor + 1$.

The balance testability of MSL and bridging faults can be studied in a similar fashion. Figure 5 illustrates the notation used for analyzing double stuck-line faults. Let $p_1, p_2, \ldots, p_k$ be lines in the circuit such that $p_1 = f_1(X_1)$, $p_2 = f_2(X_2), \ldots, p_k = f_k(X_k)$. We can write $F(X) = g(f_1(X_1), f_2(X_2), \ldots, f_k(X_k), X_{k+1})$, where $X_1 \cup X_2 \cup \cdots \cup X_{k+1} = X$. The balance testability of the MSL fault $\{p_1$ s-a-d_1, p_2 s-a-$d_2, \ldots, p_k$ s-a-$d_k\}$ ($d_1, d_2, \ldots, d_k \in \{0, 1\}$) can then be expressed in terms of the parameter $k^g_{p_1^{d_1}, p_2^{d_2}, \ldots, p_k^{d_k}}$, where $k^g_{p_1^{d_1}, p_2^{d_2}, \ldots, p_k^{d_k}}$ is the number of true vectors of g with $p_i = d_i$, $1 \leq i \leq k$. For example, the following theorem characterizes the balance redundancy of a special class of MSL faults.

Theorem 4. *Let $F(X) = g(f_1(X_1), f_2(X_2), \ldots, f_k(X_k), X_{k+1})$ be the function realized by the CUT. Let $X_1, X_2, \ldots, X_k$ be pairwise disjoint, $|X_i| = m_i$ for $1 \leq i \leq k$, $|X_{k+1}| = n$, and $p_i = f_i(X_i)$ for $1 \leq i \leq k$. Let δ be the balance offset of F,* *and let* $|\cup_{i=1}^{k}(X_i \cap X_{k+1})| = \sum_{i=1}^{k} |X_i \cap X_{k+1}| = w$. *The MSL fault* $\{p_1$ *s-a-*d_1, p_2 *s-a-*$d_2, \ldots, p_k$ *s-a-*$d_k\}$ *is balance-redundant if and only if* $k^g_{p_1^{d_1}, p_2^{d_2}, \ldots, p_k^{d_k}} = 2^{n-1} + \frac{\delta}{2^{m_1+\cdots+m_k-w}}$.

Proof: We prove the theorem for the double fault $\{p_1$ s-a-0, p_2 s-a-0$\}$, i.e., $d_1 = d_2 = 0$. Let $X_1 \cap X_3 = \{x_1, x_2, \ldots, x_u\}$ and $X_2 \cap X_3 = \{x_{u+1}, x_{u+2}, \ldots, x_w\}$. Let $\psi(p_1^{d_1}, p_2^{d_2})$ be the number of true vectors of F arising from the true vectors of g where $p_1 = d_1$ and $p_2 = d_2$. Then the number of true vectors of F is given by $|F|^1 = \sum_{d_1,d_2} \psi(p_1^{d_1}, p_2^{d_2})$, where

$$\psi(p_1^{d_1}, p_2^{d_2}) = \sum_{r_1,\ldots,r_w} k^g_{p_1^{d_1}, p_2^{d_2}, x_1^{r_1}, \ldots, x_w^{r_w}} \cdot k^{f_1^{d_1}}_{x_1^{r_1}, \ldots, x_u^{r_u}} \cdot k^{f_2^{d_2}}_{x_{u+1}^{r_{u+1}}, \ldots, x_w^{r_w}}$$

and $r_1, r_2, \ldots, r_w \in \{0, 1\}$. In the presence of the fault $\{p_1$ s-a-0, p_2 s-a-0$\}$, $|f_1|^1 = |f_2|^1 = 0$, and $|f_1|^0 = 2^{m_1}$, $|f_2|^0 = 2^{m_2}$. Therefore, for all $r_1, r_2, \ldots, r_w$, $k^{f_1}_{x_1^{r_1}, \ldots, x_u^{r_u}} = k^{f_2}_{x_{u+1}^{r_{u+1}}, \ldots, x_w^{r_w}} = 0$, and $k^{f_1'}_{x_1^{r_1}, \ldots, x_u^{r_u}} = 2^{m_1-u}$, $k^{f_2'}_{x_{u+1}^{r_{u+1}}, \ldots, x_w^{r_w}} = 2^{m_2-w+u}$. The number of true vectors in the faulty function $F^\star$ is given by $|F^\star|^1 = \psi(p_1', p_2')$, which implies that $|F^\star = 2^{m_1+m_2-w} \cdot k^g_{p_1', p_2'}$. The fault is balance-redundant if and only if $|F|^1 = |F^\star|^1$, which implies that $k^g_{p_1', p_2'} = 2^{n-1} + \frac{\delta}{2^{m_1+m_2-w}}$. ■

For example, in the circuit of Fig. 6, $z = F(a, b, c, d) = ab + cd + a'b'c'd'$, $p_1 = ab$, $p_2 = cd$, $n = 4$, and $g(p_1, p_2, a, b, c, d) = p_1 + p_2 + a'b'c'd'$. Since $k^g_{p_1', p_2'} = 1$, $\delta = 0$, and $2^{n-1} = 8$, the fault $\{p_1$ s-a-0, p_2 s-a-0$\}$ is balance-testable. For an MSL fault to be balance-redundant, $|\delta| \geq 2^{m_1+\cdots+m_k-l}$, and if $m_1+m_2 \neq l$, then $|\delta|$ must be a power of 2. This implies that in fanout-free circuits, all MSL faults are balance-testable. In other words, MSL faults in fanout-free circuits are syndrome-testable, a fact not recognized in [4]. The general conditions for balance redundancy of MSL faults are very strict, hence like SSL faults, most MSL faults can be expected to be balance-testable.

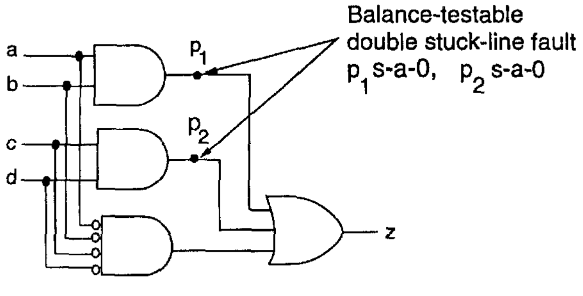

Figure 6. A balance-testable double stuck-line fault.

Defects that manifest themselves as shorts between unconnected components in circuits are called bridging faults [1]. If the affected signal lines x and y are effectively AND-ed, the fault is referred to as an AND *bridging fault*, denoted by AND(x, y). The OR *bridging fault*, OR(x, y) is defined similarly. A bridging fault can create a feedback loop, and thus transform a combinational circuit to a sequential one. The following result characterizes the balance testability of non-feedback bridging faults.

Theorem 5. *Let* $F(x_1, \ldots, x_n)$ *be the function with balance offset* δ *realized by the CUT. The following statements are logically equivalent:*

1. *The AND bridging fault between the primary inputs* x_i *and* x_j *is balance-redundant.*
2. $3k^F_{x_i', x_j'} + k^F_{x_i, x_j} = 2^{n-1} + \delta$.
3. $k^F_{x_i', x_j'}$ *is the arithmetic mean of* $k^F_{x_i', x_j}$ *and* $k^F_{x_i, x_j'}$, *i.e.* $k^F_{x_i', x_j'} = \frac{k^F_{x_i', x_j} + k^F_{x_i, x_j'}}{2}$.

Proof: Without loss of generality, let us consider the inputs x_1 and x_2. Shannon's expansion theorem implies $F(x_1, x_2, \ldots, x_n) = x_1 F(1, x_2, \ldots, x_n) + x_1' F(0, x_2, \ldots, x_n) = x_1 x_2 F(1, 1, x_3, \ldots, x_n) + x_1 x_2' F(1, 0, x_3, \ldots, x_n) + x_1' x_2 F(0, 1, x_3, \ldots, x_n) + x_1' x_2' F(0, 0, x_3, \ldots, x_n)$. Consider the AND bridging fault AND(x_1, x_2). Then $F(0, 1, x_3, \ldots, x_n) = F(1, 0, x_3, \ldots, x_n) = F(0, 0, x_3, \ldots, x_n)$ and the faulty function $F^\star(x_1, x_2, \ldots, x_n) = (x_1' + x_2') F(0, 0, x_3, \ldots, x_n) + x_1 x_2 F(1, 1, x_3, \ldots, x_n)$. Therefore, $|F^\star|^1 = 3k^F_{x_1, x_2} + k^F_{x_1', x_2'}$. This establishes the equivalence between statements 1 and 2. Statement 3 follows from the equation $k^F_{x_1', x_2'} + k^F_{x_1', x_2} + k^F_{x_1, x_2'} + k^F_{x_1, x_2} = |F|^1$. ■

Theorem 5 is valid for the OR bridging fault between primary inputs x_i and x_j if $k^F_{x_i, x_j}$ and $k^F_{x_i', x_j'}$ are interchanged in the theorem statement.

Corollary 4. *Let* $F(x_1, \ldots, x_n)$ *be the function realized by the CUT. If either* $k^F_{x_i', x_j}$ *or* $k^F_{x_i, x_j'}$ *(but not both) is odd, the AND and OR bridging faults between primary inputs* x_i *and* x_j *are balance-testable.*

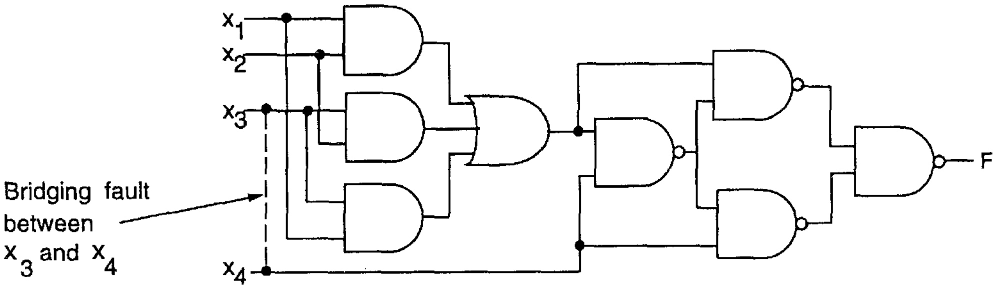

Figure 7. An example of a non-feedback bridging fault.

For example, in the circuit of Fig. 7, $k^F_{x_3,x_4} = |F(x_1, x_2, 1, 1)|^1 = 1$, $k^F_{x_3',x_4'} = |F(x_1, x_2, 0, 0)|^1 = 1$, and $\delta = 0$. Therefore, Theorems 5 implies that the faults AND(x_3, x_4) and OR(x_3, x_4) are balance-testable.

4. Design for Balance Testability

A circuit is balance-testable if it has no balance-redundant faults. In this section, we present design for balance testability (DFBT), a systematic design-for-testability technique for ensuring 100% fault coverage for SSL faults. DFBT offers several advantages over many other DFT techniques. It requires only one additional control input, and the method can be easily integrated into standard CAD tools. It does not require redesign of the underlying circuit or impose any stringent design rules. There is very little adverse effect on the normal performance of the circuit. This is in contrast to existing DFT techniques such as test-point placement [23], which require redesign and increase the circuit delay.

The basic idea of DFBT is to add one extra input and a few extra gates to the circuit under test C, as shown in Fig. 8. The AND gates are chosen to eliminate as many of the balance-redundant faults as possible in C. We first identify the balance-redundant faults, which can be done by exhaustive simulation. For this purpose, we can use a fault simulation program that computes the balance signature for each fault in C. For the time being, we assume exhaustive balance testing. We extend the method to nonexhaustive balance testing in Section 5.

We define a *redundancy cover* as a set of false vectors of F (input combinations for which F is 0), where C realizes the function $F(X)$, such that for every fault p_i s-a-d, there is at least one vector that is a test for it. If the redundancy cover consists of k false vectors of F, let $\Phi = \{\phi_1(X), \phi_2(X), \ldots, \phi_k(X)\}$, where $\phi_i(X) = x_1^{d_1} x_2^{d_2} \cdots x_N^{d_N}$ and $x_j = d_j$ $(1 \leq j \leq N)$ in the ith vector of the redundancy cover. Every detectable but balance-redundant fault is included in the redundancy cover, a consequence of the following lemma.

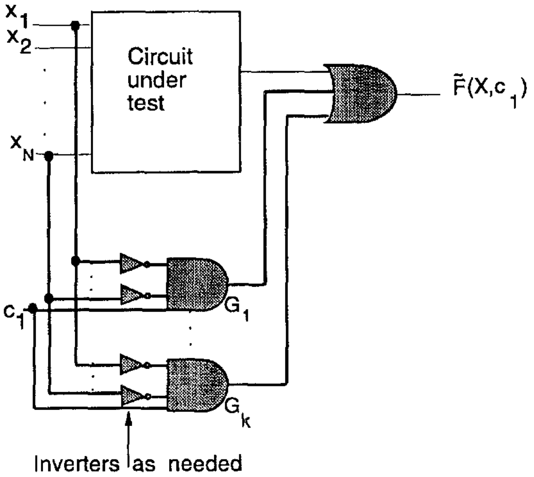

Figure 8. The general DFBT technique.

Lemma 2. *Let p s-a-1 (p s-a-0) be balance-redundant in a circuit that realizes $F(X)$. Then there exists at least one false vector V of F such that V is a test for p s-a-1 (p s-a-0).*

Proof: Let p s-a-1 be balance-redundant, and let the fault change F to $F^\star$. Suppose $F(V^0) = F^\star(V^0)$ for all false vectors V^0 of F. This implies that $F(V^1) = F^\star(V^1)$ for all true vectors V^1 of F since the fault is balance-redundant. Consequently, no input vector V exists that makes $F^\star(V) \neq F(V)$, implying that p s-a-1 is undetectable, a contradiction. Therefore, there exists a false vector V^0 which is a test for the fault. The proof for the fault p s-a-0 is similar. ■

The redundancy cover can be generated from a *balance redundancy table*, whose columns denote the false vectors $V_1, V_2, \ldots, V_k$ of F and whose rows denote the balance-redundant faults $f_1, f_2, \ldots, f_m$. Each fault f_i corresponds to a line p_{k_i} s-a-d. The entry in the ith row and jth column is 1 if the input combination V_j is a test for p_{k_i} s-a-d. In practice, it suffices to consider a small number $w << |F|^0$ of false vectors of F because there are typically only a few balance-redundant faults.

We make C balance-testable by adding a control input c_1 such that the modified circuit $\tilde{C}$ realizes the function $\tilde{F}(X) = \tilde{F}(X, c_1)$, where $\tilde{F}(X, c_1) = F(X) + c_1\phi_1(X) + c_1\phi_2(X) + \cdots + c_1\phi_k(X)$; see Fig. 8. The various ϕ_i's are obtained from the redundancy cover described above. We show later that k is small—just one or two—for common circuits. The modified circuit operates in two modes. In the normal mode, the input c_1 is set to 0, so the CUT realizes the function F. In the test mode, c_1 is first set to 0 and all combinations of X are applied to the circuit. This detects all the balance-testable faults in C. Next, c_1 is treated as an independent primary input and all combinations of c and X are applied to the circuit. This detects the faults that are balance-redundant in C.

The various steps in the DFBT procedure are listed below.

DFBT Procedure. Given a circuit C realizing the function $F(x_1, x_2, \ldots, x_N)$, construct $\tilde{C}$ from C as follows:

1. Identify the balance-redundant faults.
2. Generate a redundancy cover for the balance-redundant faults; let its size be k. Let $\phi_i(X) = x_1^{d_1}x_2^{d_2}\cdots x_N^{d_N}$, where $x_j = d_j$ in the ith vector of the redundancy cover.
3. Introduce a control input line c_1.
4. Insert $k(N+1)$-input AND gates $G_1, G_2, \ldots, G_k$ to realize the functions $c_1\phi_1(X)$, $c_1\phi_2(X), \ldots, c_1\phi_k(X)$. If necessary to meet fan-in constraints, implement G_i as a tree of AND gates.
5. Insert a $(k+1)$-input OR gate G_{k+1}, whose inputs are $F(X)$ and the outputs of $G_1, G_2, \ldots, G_k$. Make G_{k+1}'s output the primary output of $\tilde{C}$.

Although the DFBT procedure requires a significant amount of computation, this has to be carried out only once during the design process. In Section 5, we demonstrate that the procedure is computationally feasible for all the ISCAS 85 benchmark circuits. The next result shows that DFBT guarantees the detection of all (detectable) faults.

Theorem 6. *If an SSL fault is balance-redundant in C, then the fault is balance-testable in $\tilde{C}$.*

Proof: Let the fault p s-a-1 be balance-redundant in C. Then from Theorem 1, $k_p^g = 2^{n-1} + \delta/2^{m-w}$ using the notation of Fig. 3. Let u (v) be the number of ϕ_i's such that $\phi_i(X) = 1$ makes $p = 0\,(1)$. Now, $k_p^{\bar{g}} = 2^{N-n+1}k_p^g + v = 2^N + 2\delta + v$. For the fault to be balance-redundant in $\tilde{C}$, $k_p^{\bar{g}} = 2^N + \tilde{\delta}$, where $\tilde{\delta} = 2\delta + u + v$. Since the fault is included in the redundancy cover (Lemma 2), $v \geq 1$ and $k_p^{\bar{g}} \neq 2^N + \tilde{\delta}$; hence, it is balance-testable. ■

The value of k obtained from the DFBT procedure is a maximum when every balance-redundant fault requires a different AND gate. However, for many useful circuits, k is very small, as we will see in Section 4. For example, $k = 2$ for the 74283 carry-lookahead adder, which has an exceptionally large number of balance-redundant faults.

Every irredundant two-level AND-OR circuit can be made syndrome-testable by adding control inputs to the AND gates [4]. It is interesting to compare DFBT to this method for making an AND-OR circuit syndrome-testable. The circuit, shown in Fig. 9(a), which is taken from [4], realizes the function $F(x_1, x_2, x_3, x_4, x_5) = x_1x_2' + x_1'x_3 + x_2x_3' + x_4x_5 + x_4'x_5'$. The syndrome-testable circuit shown in Fig. 9(b), also taken from [4], requires two control inputs. The balance-testable version of the same circuit shown in Fig. 9(c) requires an extra gate, but only one control input. The gates that are added or modified are shaded, and additional connections are bold in Fig. 9.

The syndrome-testable design suffers from several drawbacks: (i) it requires extensive redesign of the original circuit, (ii) it is not applicable to multi-level circuits, and (iii) there is no limit to the number of additional control inputs. DFBT avoids all these drawbacks for a modest increase in chip area. Moreover, the area overhead penalty as a fraction of circuit area tends to decrease with circuit size.

For a multiple-output circuit, the DFBT procedure can be applied separately to each single-output subcircuit that contains balance-redundant faults. The additional AND gates can be shared between the different primary outputs to reduce the overhead. To illustrate this, we apply DFBT to two representative logic circuits—the 74283 carry-lookahead adder, and

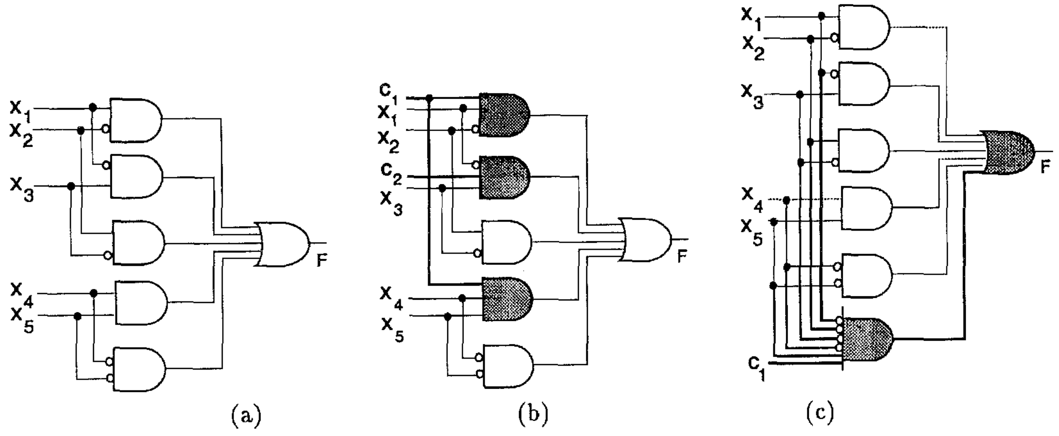

Figure 9. (a) An example circuit from [22]: (b) syndrome-testable design [22] (c) balance-testable design.

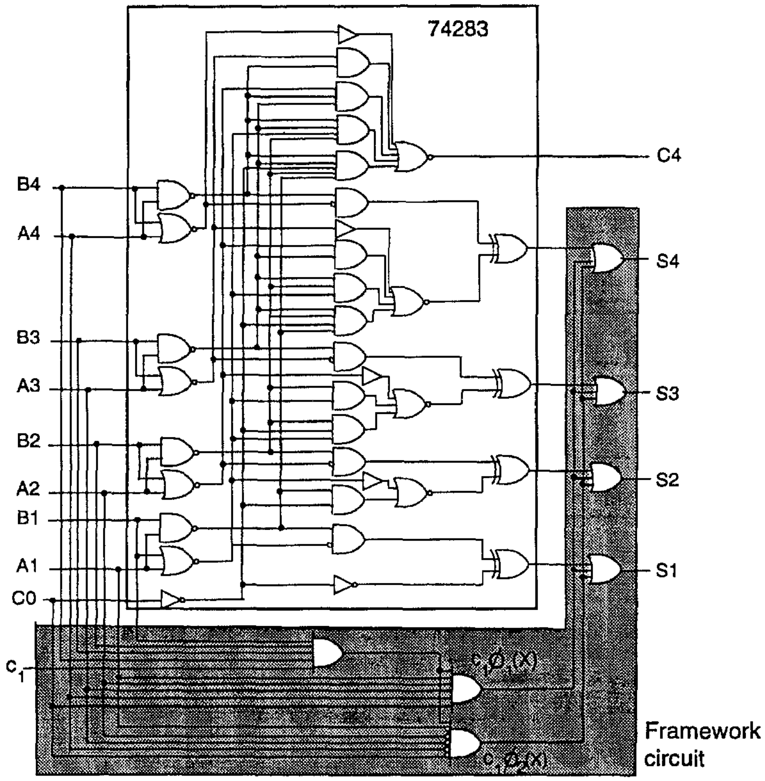

Figure 10. DFBT version of the 74283 carry-lookahead adder.

the 74181 ALU [21]. The 74181 contains only a small number (8 out of 384) balance-redundant faults (see Table 1). The 74283 carry-lookahead adder, however, contains an unusually large number of balance-redundant faults (108 out of 250). To estimate the area overhead, we count the number of extra gates weighted by their fan-in.

The 74283 circuit has five outputs $S1$, $S2$, $S3$, $S4$, and $C4$. The subcircuit feeding the carry output $C4$ is fully balance-testable, therefore the DFBT procedure has to be applied only to the four sum outputs. The resulting balance-testable circuit is shown in Fig. 10. The redundancy cover for this circuit, produced by DFBT, consists of two input vectors

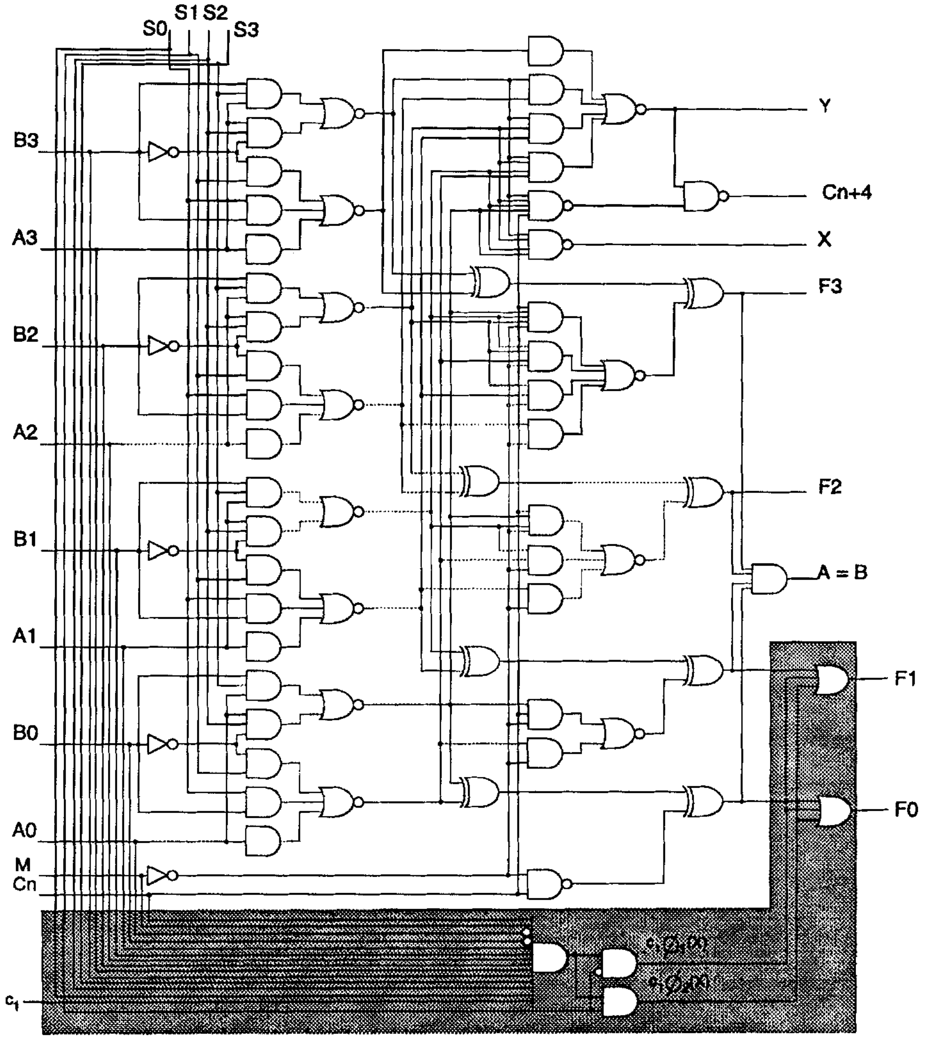

Figure 11. DFBT version of the 74181 ALU.

($k = 2$), and therefore only two AND functions $c_1\phi_1(X) = c_1{\cdot}A1{\cdot}A2{\cdot}A3{\cdot}A4{\cdot}B1{\cdot}B2{\cdot}B3{\cdot}B4{\cdot}C0$ and $c_1\phi_2(X) = c_1{\cdot}A1'{\cdot}A2'{\cdot}A3'{\cdot}A4'{\cdot}B1{\cdot}B2{\cdot}B3{\cdot}B4{\cdot}C0'$ are necessary. These functions, shown in the figure, are realized using three AND gates by factoring out the subfunction $B1{\cdot}B2{\cdot}B3{\cdot}B4$ common to $\phi_1(X)$ and $\phi_2(X)$.

The above example illustrates one of the major advantages of the DFBT technique. The logic modification is done only at the primary outputs, instead of at internal points. Thus, the DFBT procedure can be viewed as the addition of a framework circuit (shown shaded in Fig. 10) that guarantees full balance testability. The area overhead is 25% for the 4-bit adder of Fig. 10, but drops significantly for larger adder circuits. For example, the overhead is only about 15% for a 16-bit carry-lookahead adder made from 4-bit adder slices and a 4-bit carry-lookahead generator.

The 74181 ALU has eight outputs: $F0$, $F1$, $F2$, $F3$, X, Y, C_{n+4}, $A = B$. The subcircuits corresponding to the outputs $F2$, $F3$, X, C_{n+4}, $A = B$ are balance-testable, so the DFBT procedure has to be applied only to $F0$ and $F1$. The value of k for this circuit is 2, and 100% SSL fault coverage is obtained with an area overhead of 12%. Once again, the functions $\phi_1(X)$ and $\phi_2(X)$ in Fig. 9 are realized using three AND gates by factoring out a common subfunction.

The DFBT procedure can be easily extended to nonexhaustive test sets—both reduced and pseudorandom. Given test set T, we define the redundancy cover as a set of test patterns from T which (i) together detect all the balance-redundant faults, and (ii) are all either true vectors or false vectors of the function realized by the circuit under test. This allows us to use the DFBT procedure for any given test set. Lemma 2 and Theorem 6, which guarantee the completeness of the DFBT procedure, apply to (pseudo) exhaustive test sets. Therefore, it is possible that not every balance-redundant fault is included in the redundancy cover if nonexhaustive testing is used. However, we have performed extensive experiments with the ISCAS 85 benchmark circuits using reduced test sets and obtained 100% fault coverage in all cases, as we will see in the following section.

5. Experimental Results

Next, we consider the application of balance testing to large combinational circuits. Since most circuits have more than one output, we need to extend balance testing to multiple-output circuits. First, we consider a method that trades off testing time for hardware overhead. We then describe balance testing using a combination of reduced test sets and space compaction, and present experimental results for the ISCAS 85 benchmark circuits. These results show that balance testing provides very high fault coverage for large multiple-output circuits.

Balance testing can be applied to multiple-output circuits by time multiplexing the test responses from the different primary outputs. In this method, the testing process is repeated for every primary output, and a multiplexer is used to select the primary outputs, one at a time. In order to determine the fault coverage for balance testing with time multiplexing of output responses, we performed a set of simulation experiments with the ISCAS 85 benchmark circuits [20]. In these experiments, we employed reduced test sets generated by the COMPACTEST [24] and ATALANTA [25] test generation programs and explicitly generated the fault dictionaries. Tables 2(a) and 2(b) list the fault coverages obtained in these two sets of experiments.

Table 2. Fault coverage obtained for balance testing with time multiplexing of output responses for the ISCAS 85 circuits with reduced tests generated by (a) COMPACTEST, and (b) ATALANTA.

ISCAS 85 benchmark circuit	No. of test patterns	No. of detectable faults	No. of faults detected	Percentage fault coverage
		(a)		
c432	48	520	485	93.27
c499	59	750	708	94.40
c880	30	942	938	99.36
c1355	95	1566	1527	97.51
c1908	129	1870	1835	98.13
c2670	75	2630	2528	96.12
c3540	113	3287	3158	96.08
c5315	59	5291	4913	92.06
c6288	23	7710	7707	99.97
c7552	88	7419	7160	96.51
		(b)		
c432	61	520	509	97.88
c499	63	750	709	94.53
c880	66	942	929	98.62
c1355	87	1566	1545	98.66
c1908	127	1870	1848	98.82
c2670	121	2630	2548	96.88
c3540	174	3287	3164	96.26
c5315	144	5291	5221	98.68
c6288	40	7710	7660	99.35
c7552	236	7419	7151	96.39

The above technique for time multiplexing the test responses requires only one response compression counter, but it suffers from the drawback that a k-output circuit increases the test application time by a factor of k. We next describe a method for merging the k test response streams into a single-bit stream, which can then be compressed into a short signature using balance testing. Such two-stage compression of the test response often requires less logic then traditional compression techniques such as multiple-input signature registers (MISRs) [18].

The k-bit data stream from a k-output circuit can be compressed to a 1-bit data stream by a parity tree (see Fig. 12). The use of parity trees for space compaction was proposed in [26], and experimental results for the ISCAS 85 circuits presented in [27] and [17] indicate that parity compression introduces very little fault masking. Moreover, fault masking can be eliminated altogether either by suitably choosing the test set or by modifying the circuit under test [17]. Therefore, we can keep the test application time low without requiring additional response compression counters.

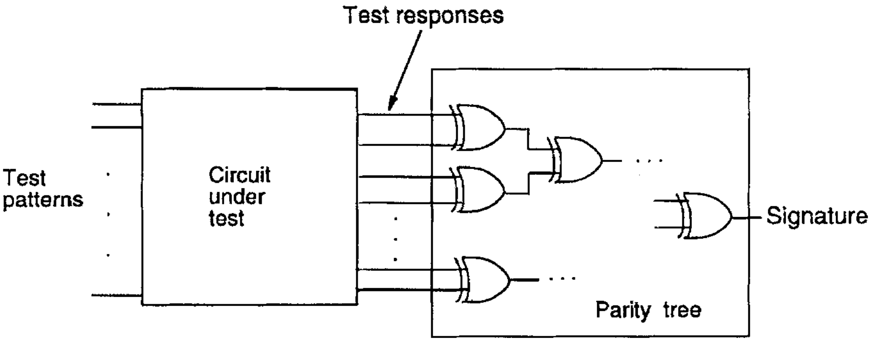

Figure 12. Merging of k test response streams by means of a parity tree.

We next determined the fault coverage of balance testing for the benchmark circuits using a parity tree to merge the output responses into a single-bit stream.

Table 3. Fault coverage of balance testing for the ISCAS 85 circuits with space compaction of output responses and reduced test sets generated by (a) COMPACTEST, and (b) ATALANTA. (The number of detectable faults represent the faults that are not masked by the parity tree.)

ISCAS 85 benchmark circuit	No. of test patterns	No. of detectable faults	No. of faults detected	Percentage fault coverage
		(a)		
c432	48	514	513	99.81
c499	59	687	687	100
c880	30	942	942	100
c1355	95	1558	1549	99.42
c1908	129	1856	1840	99.14
c2670	75	2549	2434	95.49
c3540	113	3210	3210	100
c5315	59	5204	5179	99.52
c6288	23	7605	7563	99.45
c7552	88	7361	7306	99.25
		(b)		
c432	61	517	517	100
c499	63	749	726	96.79
c880	66	942	940	99.79
c1355	87	1559	1550	98.97
c1908	127	1863	1855	99.19
c2670	121	2623	2595	98.67
c3540	174	3270	3211	97.68
c5315	144	5288	5288	100
c6288	40	7701	7673	99.52
c7552	236	7370	7316	99.27

The experiments were performed with reduced deterministic test sets generated by COMPACTEST and ATALANTA. Table 3 lists the fault coverage obtained with this method. Note that we have not considered the faults that are masked by the parity tree; as stated earlier, fault masking in the parity tree can be easily avoided.

From Table 3, we see that very high post-compaction fault coverage (over 99%) is achieved with balance testing using test patterns produced by standard test generation programs. We can increase the coverage to 100% by using a test pattern generator tailored to balance testing. We can also achieve 100% coverage for all the circuits by applying DFBT with the standard test sets. The redundancy cover for each circuit consists of either one or two input vectors, implying low hardware overhead. This is a direct consequence of the fact that there are so few balance-redundant faults.

6. Conclusion

Balance testing is a promising approach to the built-in self-testing of logic circuits. A key advantage of balance testing is that the testability of various fault types can be directly analyzed. We derived necessary and sufficient conditions here for the detection of SSL faults. These conditions are very strict, implying that in almost all practical cases, complete fault detection can be guaranteed. Another advantage of balance testing is that the fault-free signature is always zero, therefore there is no need to store a precomputed reference signature.

Design for balance testability (DFBT) provides a systematic method for always achieving 100% fault coverage with balance testing. DFBT does not require redesign of the CUT or impose stringent design rules. It effectively places the circuit under test in a low-cost framework circuit that ensures full testability. This framework can be added automatically to any design and can be used with any complete test set. Balance testing can be implemented for large circuits by time multiplexing the output responses, or by using a parity tree to merge the responses. We are currently studying the application of balance testing to general sequential circuits, and investigating ways to reduce the hardware overhead.

Acknowledgments

We would like to thank Professors Sudhakar Reddy and Irith Pomeranz of the University of Iowa for providing us with COMPACTEST. We are also grateful to Professor Dong Ha of Virginia Tech. for giving us ATALANTA. Stimulating discussions with Mark Hansen of General Motors Corporation are gratefully acknowledged.

References

1. M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, New York, 1990.
2. J.P. Hayes, "Check Sum Methods for Test Data Compression," *Journal of Design Automation and Fault-Tolerant Computing*, vol. 1, pp. 3–7, January 1976.
3. A. Tzidon, I. Berger, and M. Yoeli, "A Practical Approach to Fault Detection in Combinational Circuits," *IEEE Transactions on Computers*, vol. C-27, pp. 968–971, October 1978.
4. J. Savir, "Syndrome-Testable Design of Combinational Circuits," *IEEE Transactions on Computers*, vol. C-29, pp. 442–451, June 1980.
5. R.A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," *Hewlett-Packard Journal*, vol. 28, pp. 2–8, September 1977.
6. K. Chakrabarty, *Balanced Boolean Functions*, M.S. thesis, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, September 1992; also published as Technical Report CSE-TR-140-92.
7. P. Agrawal and V.D. Agrawal, "Probabilistic Analysis of Random Test Generation Method for Irredundant Combinational Networks," *IEEE Transactions on Computers*, vol. C-24, pp. 691–694, July 1975.
8. J.P. Hayes, "A NAND Model for Fault Diagnosis in Combinational Logic Networks," *IEEE Transactions on Computers*, vol. C-20, pp. 1491–1506, December 1971.
9. W.H. Kautz, "State-Logic Relations in Autonomous Sequential Networks," *Proc. 1958 Eastern Joint Computer Conference*, 1958, pp. 119–127.
10. B. Elspas, "Self-Complementary Symmetry Types of Boolean Functions," *IRE Transactions on Electronic Computers*, vol. EC-9, pp. 264–266, June 1960.
11. A. Chatterjee and J.A. Abraham, "Test generation, Design-for-Testability and Built-In Self-Test for Arithmetic Units Based on Graph Labeling," *Journal of Electronic Testing: Theory and Applications*, vol. 2, pp. 351–372, December 1991.
12. S. Kundu, "Basis Sets for Synthesis of Switching Functions," *IEEE Transactions on Computers*, vol. 41, pp. 489–493, April 1992.
13. S.B. Akers, "A Parity Bit Signature for Exhaustive Testing," *IEEE Transactions on Computer-Aided Design*, vol. 7, pp. 333–338, March 1988.
14. C.-I.H. Chen and J.T. Yuen, "Automated Synthesis of Pseudo-Exhaustive Test Generator in VLSI BIST Design. BIST Design," *IEEE Transactions on VLSI Systems*, vol. 3, pp. 273–291, September 1994.
15. D. Kagaris, F. Makedon, and S. Tragoudas, "A Method for Pseudo-Exhaustive Test Pattern Generation," *IEEE Transactions on Computer-Aided Design*, vol. 13, pp. 1170–1178, September 1994.
16. R. Srinivasan, S.K. Gupta, and M.A. Breuer, "An Efficient Partitioning Strategy for Pseudo-Exhaustive Testing," *Proc. 1993 Design Automation Conference*, 1993, pp. 242–248.
17. K. Chakrabarty and J.P. Hayes, "Efficient Test Response Compression for Multiple-Output Circuits," *Proc. 1994 Int. Test Conference*, 1994, pp. 501–510.
18. Y. Zorian and A. Ivanov, "Programmable Space Compaction for BIST," *Proc. 1993 Int. Symp. on Fault-Tolerant Computing*, 1993, pp. 340–349.
19. Y. You and J.P. Hayes, "Implementation of VLSI Self-Testing by Regularization," *IEEE Transactions on Computer-Aided Design*, vol. 8, pp. 1261–1271, December 1988.
20. F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Simulator in Fortran," *Proc. 1985 Int. Symp. on Circuits and Systems*, 1985, pp. 695–698.
21. Texas Instruments, *The TTL Data Book*, vol. 2. Dallas, 1988.
22. W.H. McAnney and J. Savir, "Built-in Checking of the Correct Self-Test Signature," *IEEE Transactions on Computers*, vol. 37, pp. 1142–1145, September 1988.
23. J.P. Hayes and A.D. Friedman, "Test Point Placement to Simplify Fault Detection," *IEEE Transactions on Computers*, vol. C-33, pp. 727–735, July 1974.
24. I. Pomeranz, L.N. Reddy, and S.M. Reddy, "COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits," *Proc. 1991 Int. Test Conference*, 1991, pp. 194–203.
25. H.K. Lee and D.S. Ha. On the Generation of Test Patterns for Combinational Circuits. Technical Report No. 12–93, Dept. of Electrical Eng., Virginia Polytechnic Institute and State University.
26. S.M. Reddy, K.K. Saluja, and M.G. Karpovsky, "A Data Compression Technique for Built-in Self-Test," *IEEE Transactions on Computers*, vol. C-37, pp. 1151–1156, September 1988.
27. H. Fujiwara and A. Yamamoto, "Parity-Scan Design to Reduce the Cost of Test Application," *IEEE Transactions on Computer-Aided Design*, vol. 12, pp. 1604–1611, October 1993.

Krishnendu Chakrabarty is Assistant Professor of Electrical, Computer, and Systems Engineering at Boston University. He received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively, all in Computer Science and Engineering. While at the University of Michigan, he was a research assistant at the Advanced Computer Architecture Laboratory in the Department of Electrical Engineering and Computer Science. Dr Chakrabarty's research interests are in computer-aided design of VLSI circuits and systems, testing, design verification, and fault-tolerant computing. He is a member of IEEE, ACM, and an associate member of Sigma Xi.

John P. Hayes is Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, where he teaches and does research in the areas of computer architecture; computer-aided design, verification and testing; VLSI design; and fault-tolerant embedded systems. He received the B.E. degree from the National University of Ireland, Dublin, in 1965, and the M.S. and Ph.D. degrees from the University of Illinois, Urbana-Champaign, in 1967 and 1970, respectively, all in electrical engineering. While at the University of Illinois, he participated in the design of the ILLIAC III computer. In 1970 he joined the Operations Research Group at the Shell Benelux Computing Center in The Hague, where he worked on mathematical programming. From 1972 to 1982 Dr. Hayes was a faculty member of the Departments of Electrical Engineering-Systems and Computer Science of the University of Southern California, Los Angeles. He joined the University of Michigan in 1982. He was the founding director of the University of Michigan's Advanced Computer Architecture Laboratory. He was Technical Program Chairman of the 1977 International Conference on Fault-Tolerant Computing, Los Angeles, and the 1991 International Computer Architecture Symposium, Toronto. Dr. Hayes is the author of numerous technical papers and five books, including Computer Architecture and Organization, (2nd ed., McGraw-Hill, 1988), Hierarchical Modeling for VLSI Circuit Testing, (Kluwer, 1990; coauthored with D. Bhattacharya), and Introduction to Digital Logic Design, (Addison-Wesley, 1993). He has served as editor of various technical journals, including the IEEE Transactions on Parallel and Distributed Systems and the Journal of Electronic Testing. Dr. Hayes is a Fellow of IEEE and a member of ACM and Sigma Xi.