Design of TSC Checkers for Implementation in CMOS Technology

Sandip Kundu  
IBM T. J. Watson Research Center  
PO Box 218, Yorktown Heights  
NY 10598, USA

Sudhakar M. Reddy  
Department of Elect. & Comp. Engineering  
University of Iowa, Iowa City  
IA 52242, USA

Abstract

Fault model for CMOS digital circuits comprise of FET stuck-open and FET stuck-on faults, in addition to line stuck-at faults used conventionally. It has been shown that delays in stuck-open and FET stuck-on faults, in addition to line stuck-at faults, may not be self-testing for CMOS fault model, violating one of the conditions of Totally Self-Checking circuits. In this paper we suggest a design procedure that constructs a self-testing circuit for CMOS fault model using at most four levels. Thus the previous designs could be adapted for CMOS fault model without any penalty. The new design also makes it possible to meet arbitrary fan-in restrictions.

I. Introduction

Testing is needed to insure reliability of VLSI chips. On-line testing allows us to monitor faults even when an application is running. CMOS has emerged to be the dominant technology for manufacturing digital logic circuits [1]. Classical fault model consisting of circuit lines stuck-at 0 or 1 faults was shown to be inadequate for CMOS circuits [2,3]. Transistor stuck-on (TSON) and transistor stuck-open (TSOP) faults were added to this model to capture the effects of physical faults [2]. Testing TSOP faults in a static CMOS combinational logic circuit requires two-pattern testing [4,5]. A two-pattern test \(<T_1, T_2>\) consists of a sequence of two inputs, the first one of which is called the initializing input \(T_1\) and the latter one is called the test input \(T_2\).

In two-pattern testing environment when an initializing input is changed to a test input, transients may be produced in circuit lines. In TSOP fault testing in CMOS circuits some of these transient signal values may undo the effect of the initializing input causing a test invalidation problem [6]. To circumvent test invalidation problem, tests called robust tests were introduced [6]. However for given faults, in a circuit under test, robust tests may not exist [6,7].

Self-Checking is a general term used to describe logic networks that allow an on-line detection of hardware faults [8]. Several Self-Checking circuits have been proposed. Typically, these circuits have their outputs encoded in an error detecting code such that no fault in the circuit can modify the output to an unintended codeword. Totally Self-Checking (TSC) circuits were introduced by Carter and Schneider [9]. An extension of concept of TSC circuits was proposed by Anderson to insure correct operation until the first indication of error [10]. Figure 1 shows a general TSC system.

Under normal operation a TSC circuit receives only a subset of all possible inputs called input codes and produces a subset of all possible outputs called output codes. The following definitions are due to Anderson [8] and describes the properties of a TSC system.

**Definition 1:** A circuit is called self-testing for a fault set \(\Phi\), iff for any single fault \(\tau \in \Phi\) the circuit produces a non-code output for at least one code input.

**Definition 2:** A circuit is called code-disjoint if maps an code input to a code output and non-code input to a non-code output.

**Definition 3:** A circuit is called a TSC checker if it is both self-testing and code-disjoint.

If the input code of a TSC checker is the set of all \(m\)-out-of-\(n\) words, then it is called a \(m\)-out-of-\(n\) code checker. Conventionally a TSC checker has two outputs that produce 00 and 11 under fault-free situation, 01 and 10 under fault-free environment, 01 and 11 are the error outputs.

Fault model assumed greatly influence the design and performance of TSC circuits. Smith proposed a class of two-level \(m\)-out-of-\(n\) checkers based on line stuck-at fault model [11]. Manthani and Reddy examined Smith's TSC checkers for \(m\)-out-of-\(n\) codes under extended fault model for CMOS and found that many of these checkers do not remain TSC under extended fault model [7]. They found that in order to remain TSC under extended fault model Smith's checkers would have to satisfy additional conditions [7]. In this paper, we re-design...
Smith's two-level TSC checkers that do not satisfy Manthani and Reddy's condition(s) and derive four-level checkers that are TSC under extended fault model. The four-level circuits can further be transformed into multi-level circuits, if needed to satisfy fan-in requirements without loss of TSC property.

II. Preliminaries

In this section the previous results are summarized. First the following definitions.

Definition 4 [7]: A binary n-tuple x is said to one-cover (zero-cover) a binary n-tuple y iff Is(0) appear in every position of x where y has a 1(0). For example 110000 is a one-cover of 110000 and 110000 is a zero-cover of 1110000.

Definition 5 [7]: Let Λ be a set of n-tuples. A binary n-tuple x, is a non-redundant external (NRE) one-cover (zero-cover) with respect to Λ of an n-tuple y, y ∈ Λ, iff:

1. x is a one-cover (zero-cover) of y, and
2. x is not a one-cover of any z ∈ Λ, z ≠ y.

Definition 6 [7]: Let Λ be a set of n-tuples. A binary n-tuple x, x ∈ Λ, is a non-redundant internal (NRI) one-cover (zero-cover) with respect to Λ of an n-tuple y if:

1. x is a one-cover (zero-cover) of y, and
2. No other element z ∈ Λ one covers (zero-covers) y.

Notation

A set of all m-out-of-n codewords

x₁, x₂, ..., xₙ n checker inputs

vertex, n-cube, n-tuple

A binary word (a₁, a₂, ..., aₙ)

m-vertex

A binary word with weight m

f₁, f₂ outputs of a TSC checker

A₁ subset of A such that f₁ = 1 and f₂ = 0 for any input from A₁

A₂ subset of A such that f₁ = 0 and f₂ = 1 for any input from A₂

<T₁, T₂> a two-pattern test;

T₁ is the initializing input and

T₂ is the test input.

The design of two-level TSC checkers for constant weight codes have been studied earlier [11] and the following convention was proposed.

1. If the weight of an input vector is less than m, the outputs of the checker are both 0.
2. If the weight of an input vector is greater than m, the outputs of the checker are both 1.
3. If the weight of an input vector is equal to m, the outputs of the checker are either 10 or 01.

It is obvious from the above specification that the functions f₁ and f₂ are unate [12]. Furthermore the following Lemma results directly from the above impositions:

Lemma 1 [11]: While partitioning the set of all m-out-of-n codes A into A₁ and A₂ for realizing f₁ and f₂ respectively, the following conditions must hold true:

C₁. Each vertex with weight m+1 is a one-cover of at least one member of A₁ and one member of A₂.
C₂. Each vertex with weight m-1 is one-covered by at least one member of A₁ and one member of A₂.

Note that a circuit for f₁, f₂ may be implemented by realizing a vector in A₁ (A₂) or by realizing a 0 for every vector in A₁ (A₂). Since the functions are unate, they are realizable by inverter free circuits.

Getting satisfactory partitions {A₁, A₂} based on the above conditions are connected with Ramsey's theorem [11]. Such partitions do not exist for all values of m and n (n > m). Our construction only applies to those values of m and n for which a valid partition can be obtained. k-out-of-2k, k+1-out-of-2k+1, k-out-of-2k+1 and k+1-out-of-2k are the most important cases from the efficiency issue and methods have been found to obtain valid partitions in these cases [7]. Using only NAND and NOR gates there are three ways of realizing a two-level, 2 output TSC checker:

1. Both f₁ and f₂ are realized by NAND-NAND logic (referred to as wholly NAND-NAND).
2. Both f₁ and f₂ are realized by NOR-NOR logic (referred to as wholly NOR-NOR).
3. f₁ realized by NAND-NAND and f₂ realized by NOR-NOR logic (referred to as mixed logic).

Next we give robust tests for NAND and NOR gates. An n-input NAND gate requires n robust tests and 1 non-robust two-pattern test. T₁, for all the robust tests are identical and is given by 11...1, an all 1 input. T₂ is derived by substituting 0 for a 1 in T₁. Since T₂ has n Is, there are n T₃s and n two-pattern tests. The non-robust two-pattern test is obtained by swapping T₁ and T₂ of any of the n robust tests described above. The tests for an n-input NOR gate is similar except that T₁ for the n robust tests is an all zero pattern and T₂ is obtained by replenishing a 0 by 1.

The following three Lemmas are due to Manthani and Reddy [7].

Lemma 2 [7]: A wholly NAND-NAND realization results in a TSC checker for the extended fault model iff in addition to C₁ and C₂ the following condition is satisfied:

C₃. Each m-vertex belonging to A₁ (A₂) has an NRE (m+1) one-cover with respect to A₁ (A₂).

Lemma 3 [7]: A wholly NOR-NOR realization results in a TSC checker for the extended fault model iff in addition to C₁ and C₂ the following condition is satisfied:

C₄. Each m-vertex belonging to A₁ (A₂) is an NRI (m-1) one-cover with respect to A₁ (A₂).

Lemma 4 [7]: A mixed realization results in a TSC checker for the extended fault model iff in addition to C₁ and C₂ the following condition is satisfied:
C5. Each m-vertex belonging to $A_i$ has an NRE $(m+1)$ one-cover with respect to $A_i$ and is also an NRI $(m-1)$ one-cover with respect to $A_i$.

These Lemmas make the scope of application of two-level circuits very limited. In this paper we abandon the premise of two-level designs. This helps us design TSC checkers for all partitions that satisfy C1 and C2. C1 and C2 conditions are not due to the technology and must be satisfied by any checker. Therefore design presented here is very broad in its scope.

Definition 7: NAND, NOR, AND, OR and INV gates are called primitive gates.

All primitive have one characteristic in common. In the pFET (nFET) network of these gates, the FETs are connected either in parallel or in series but never in combination of both.

Lemma 5 [13]: In a primitive gate circuit where all single faults are robustly testable, a high fan-in primitive gate can be replaced by a functionally equivalent tree of low fan-in primitive gates and the resultant circuit remains robustly testable by the same test set as before.

Application of this Lemma makes it possible to realize TSC circuits under extended fault model that meets arbitrary fan-in restrictions.

III. TSC CMOS checker design

Design Procedure 1

Partition the set of all m-out-of-n codes $A_i$ into $A_{i1}$ and $A_{i2}$ satisfying Lemma 1. $f_i$ and $f_{i2}$ are realized by separate circuits. If $m = 1$ or $m = n - 1$ then the functions $f_i$ and $f_{i2}$ can be realized by single gates only. So, we need not be concerned about the testability problem. If $1 < m < n - 1$, then we may implement $f_i$ ($f_{i2}$) by a two-level circuit (say NAND-NAND) and check if the resultant circuit remains robustly testable by the same test set as before.

Design Procedure 2

This is the dual of Design Procedure 1. In this case $f_i$ and $f_{i2}$ can be written as:

$$f_i = (x_i + f'_{i2}) f_{i2}^{\hat{\cap}}$$
$$f_{i2} = (x_i + f'_{i1}) f_{i1}^{\hat{\cap}}$$

Realize $f'_{i1}$ and $f'_{i2}$ by NOR-NOR circuits. $f_{i1}$ and $f_{i2}$ are realized by NAND-NAND circuits.

Theorem 1: The four-level circuit derived by Design Procedure 1 and shown in Figure 2 is robustly testable with respect to all single transistor stuck-open faults.

Proof: Firstly we identify the locations where robust tests are needed. Non-robust tests can be easily derived from the stuck-at fault tests. For example the nFETs of the NAND gate that produces $f_i$ require non-robust two-pattern tests. The two patterns can be derived as <test for $f_i$ stuck at 0, test for $f_{i2}$ stuck at 0>.
stuck-at 1>. In non-robust testing we are not concerned about test invalidation. That simplifies the situation.

Referring to Figure 2, if we can demonstrate that the pFETs connected to the input lines 1, 2, 3, 4, 7, 9 and nFET connected to lines 6, 8 are robustly testable with respect to stuck-open fault then it is proved that all faults requiring robust tests are tested robustly. Lines 6, 7, 8 and 9 are arbitrary lines in the configuration. pFET connected to line 9 and nFET connected to line 8 receive primary inputs. Hence by an earlier result [6] it is always possible to find robust tests for stuck-open faults of the transistors connected to these lines.

Let X and Y refer to combination of inputs excluding x, X and Y can be described by a binary (n-1)-tuple. If x, X is a m-out-of-n code then x, = 1 implies that X is of weight (m-1) and x, = 0 implies that X is of weight m. For testing the pFET connected to line 1, find an input x, X (m-out-of-n) such that:

\[ x, = 1, f_1(X) = 1 \text{ and } f_1(Y) = 0. \]

Let this be T, our test. Switching this sequence to \(<T_2, T_1>\) tests the pFET connected to line 3 robustly for TSOP fault.

For testing the pFET connected to line 2 find T, and T, such that for T, x, = 0 and f, (X) = 0 and for T, x, = 0 but f, (Y) is 1.

To test the pFET connected to line 4, hold x, at 1 and change f, from 1 to 0. Sensitization is guaranteed because x, = 1 implies f, = 0 (otherwise it violates C2).

Now we consider testing the nFET connected to line 6. There exists a test input for line 6 stuck-at 1 (say x, X). x, for this input is 1 and the input to line 8 is 0. Use this as Tg and derive T, from T2 by changing input to line 8 to 1 and input x, to a 0.

Finally, we consider testing pFET of the gate connected to line 7. Let T, be the input that tests line 9 stuck-at 0. Derive T, from this by setting x, = 1 and input to line 9 = 0. \(<Q.E.D>\)

**Theorem 2**: The four-level circuit derived by Design Procedure_2 is robustly testable with respect to all single transistor stuck-open faults.

**Proof**: Similar to Theorem 1.

Theorem 1 and 2 guarantees that the above checkers are self-testing with respect to all transistor stuck-open faults. They are also self-testing with respect to single stuck-at faults. Since the circuits realize the same function as Smith's checkers they are code disjoint. Thus these are TSC checkers for m-out-of-n codes for extended fault model.

**IV. Summary**

In this paper we have considered the problem of tests that remain valid even in the presence of delays in CMOS logic circuits under test. Such tests were called robust tests. Then the problem of robust testable Totally Self-Checking Checkers were considered and a comprehensive solution was presented that obviates the necessity of redesigning TSC checkers for constant weight codes for extended fault model. This was accomplished by changing two-level implementations to multi-level.

**References**