Automotive Electronics Council
Component Technical Committee

ATTACHMENT 7

AEC - Q100-007 Rev-B

FAULT SIMULATION AND FAULT GRADING

Automotive Electronics Council

Component Technical Committee

Acknowledgment

Any document involving a complex technology brings together experience and skills from many sources. The Automotive Electronics Council would especially like to recognize the following significant contributors to the development of this document:

Sustaining Members:

Mark A. Kelly	Delphi Corporation
Jean Clarac	Siemens VDO
Brian Jendro	Siemens VDO
Hadi Mehrooz	Siemens VDO
Robert V. Knoell	Visteon Corporation

Associate Members:

Tim Haifley	Altera
Daniel Vanderstraeten	AMI Semiconductor
Earl Fischer	Autoliv
Mike Klucher	Cirrus Logic
Xin Miao Zhao	Cirrus Logic
John Timms	Continental Automotive Systems
Roy Ozark	Continental Automotive Systems
Nick Lycoudes	Freescale
Kenton Van Klompenberg	Gentex
Werner Kanert	Infineon Technologies
Elfriede Geyer	Infineon Technologies
John Bertaux	International Rectifier
Gary Fisher	Johnson Controls
Tom Lawler	Lattice Semiconductor
Cliff Jindra	Lattice Semiconductor
Scott Daniels	Maxim
Tom Tobin	Maxim
Mike Buzinski	Microchip
Nick Martinez	Microchip
Annette Nettles	NEC Electronics
Masamichi Murase	NEC Electronics
Zhongning Liang	NXP Semiconductors
Mark Gabrielle	ON Semiconductor
Ken Berry	Renesas Technology
Bruce Townsend	Spansion
Adam Fogle	Spansion
Brian Mielewski	STMicroelectronics
James Williams	Texas Instruments
Diana Siddall	Texas Instruments
Anca Voicu	Xilinx

Guest Members:

David Locker	AMRDEC
Jeff Jarvis	AMRDEC

Other Contributors:

Ramon Aziz	Delphi
Ken Tumin	Freescale
Matthew Stout	Freescale
Stefan Eichenberger	NXP
Amrit Vivekanand	Renesas
Nicole Cunningham	Texas Instruments
Shahin Toutounchi	Xilinx

Automotive Electronics Council

Component Technical Committee

NOTICE

AEC documents contain material that has been prepared, reviewed, and approved through the AEC Technical Committee.

AEC documents are designed to serve the automotive electronics industry through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than AEC members, whether the standard is to be used either domestically or internationally.

AEC documents are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action AEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the AEC documents. The information included in AEC documents represents a sound approach to product specification and application, principally from the automotive electronics system manufacturer viewpoint. No claims to be in conformance with this document shall be made unless all requirements stated in the document are met.

Inquiries, comments, and suggestions relative to the content of this AEC document should be addressed to the AEC Technical Committee on the link http://www.aecouncil.com.

Published by the Automotive Electronics Council.

Automotive Electronics Council
Component Technical Committee

Change Notification

The following summary details the changes incorporated into AEC-Q100-007 Rev-B:

- **Section 2, Purpose: Added new statement.**
- **Section 3, Definitions: Added new definitions section, including items 3.1 to 3.20.**
- **Figure 1, Typical Fault Simulation and Fault Grading Procedure Flow: Added new Figure illustrating the typical procedure flow for fault simulation and fault grading.**
- **Figure 2, Device Simulation: Added new figure illustrating device simulation for integrated circuits.**
- **Figures 3 and 4, Fault Types and Fault Models, respectively: Added new figures illustrating the types of faults and fault models defined in this document.**
- **Section 4.2.1.1, Stuck-at Fault Model: Added requirements for detection of stuck-at faults.**
- **Section 4.2.1.3, I_{DDQ} Pseudo Stuck-at Fault Model: Added new section; also added sub-sections 4.2.1.3.1 and 4.2.1.3.2 listing requirements for detection.**
- **Section 4.2.1.4, Transition Delay Fault Model: Added new section.**
- **Section 4.2.2, Detectable Fault Simulation for Memory: Added new section; also added sub-sections 4.2.2.1 to 4.2.2.4.**
- **Section 4.7 and Figure 6, Types of Stuck-at Faults: Added new section and figure summarizing the types of stuck-at faults defined in this document.**
- **Section 4.7.1, Untestable Faults: Provided additional information on untestable faults, specifically TYPE1 and TYPE2 faults.**
- **Section 4.7.1.1, TYPE1 Untestable Faults: Added wording, and added sections where applicable, to define the primary categories of TYPE1 untestable faults.**
- **Section 4.7.1.2, TYPE2 Untestable Faults: Added wording, and new sections where applicable, to define the primary categories of TYPE2 untestable faults.**
- **Section 4.7.2, Testable Faults: Added wording, and new sections where applicable, to define the primary categories of faults detected by implication.**
- **Section 5.3, Test Coverage: Modified equation used to calculate percent of faults detected to improve clarity. Added wording, and new sections where applicable, to better define the various aspects of test, test coverage, and coverage reporting.**
- **Section 6, Acceptance Criteria: Added wording, and new sections where applicable, to better define the qualification requirements (both analog and digital circuits), theoretical field reject rate, test sequence alterations ,and production fault coverage.**

Automotive Electronics Council
Component Technical Committee

METHOD - 007

FAULT SIMULATION AND FAULT GRADING

Text enhancements and differences made since the last revision of this document are shown as underlined areas.

1. SCOPE

This test method defines fault grading procedure and specifies a level to which the manufacturing test program for the device under test must detect faults. Parametric failures are not covered. Another term for fault grading is fault simulation. Fault grading applies to all digital circuits including the digital portion of mixed signal and linear circuits. Fault grading does not apply to the linear portion of the circuits.

Also, this document covers modeling and logic simulation requirements; the assumed fault model and fault simulation requirements; and the procedure that must be followed to evaluate and report test coverage.

2. PURPOSE

Device quality is determined by three factors:

- Quality of the fault model: *Does the fault model adequately model the effect of manufacturing defects?*
- Fault coverage: *Given a fault model, how many or the circuit primitives are tested against these faults?*
- Environmental defect activation conditions: *Some defects manifest themselves only (or more prominently) at certain activation conditions (voltage, temperature, frequency). These activation conditions hence need be reflected in the test setup to be effective.*

The purpose of this test method is to develop the optimal fault coverage of the component using current fault simulation models in order to minimize defects and report the fault coverage metric to the end user.

This test method does *not* discuss the validity of the fault models nor the proper activation conditions.

3. DEFINITIONS

3.1 Blocked Fault

A single stuck-at fault for which no test exists because the propagation path (to an observed node) is blocked.

3.2 Collapsed Fault

One single stuck-at fault for each fault equivalence grouping. The representative fault is the one at the location furthest downstream in the cells for the faults in the equivalence grouping.

3.3 Defect

A physical deformation of a circuit implementation, which makes the affected die behave differently than intended by design.

3.4 Detect by Implication

A fault that is implied to be detected by the current operation of the scan chain and chain test pattern, or through detection of a certain fault.

3.5 Detected Fault

A single stuck-at fault for which a test has been generated.

3.6 Fault

A simplified model of the behavior of possible physical defects. Common fault models include:

- Stuck-at models: A defect is modeled as tying a circuit node to either logic-1 or logic-0, independent of switching sequence.
- Sequential models: A defect is modeled as inhibiting (or delaying) a sequential transition through the affected circuitry. Examples include transition delay faults.
- Current based models (such as I_{DDQ}): A defect is modeled as having a measurable impact on the static or dynamic current consumed by the circuit.

3.7 Fault Coverage

The result of a fault grading process, where [# faults detected] / [# faults modeled] determines the level of fault coverage.

3.8 Fault Equivalence

Two single stuck-at faults are equivalent if they have exactly the same test set. It is expected that simple equivalence rules are applied across individual library cells.

3.9 Fault Grading

Simulation based process to determine the coverage of a given test set against a given fault model.

3.10 Redundant Fault

A single stuck-at fault for which no test exists and is neither blocked, tied, or unused.

3.11 Stuck-at Fault

An input or output of the library cell, or a primary input or primary output fixed at zero (0) or one (1).

3.12 Test

A sequence of stimuli and response observations designed to identify whether a circuit is fault-free.

3.13 Test Coverage

The result of a test grading process, where [# faults detected] / [(# faults modeled) - (# of undetectable faults)] determines the level of test coverage. See Section 4.8 for details.

3.14 Test Grading

A process to determine the coverage of a given test set against the set of relevant fault models and within each relevant circuit block.

3.15 Testable Fault

A single stuck-at fault for which a test exists.

3.16 Tied Fault

A single stuck-at fault for which no test exists because one or more controlling input is tied to a logical value.

3.17 Undetectable Fault

Undetectable faults are those faults which exist in the model and the actual circuit but cannot be verified by propagation to an observable output. For further details, see Section 4.7.2.4.

3.18 Undetected Fault

A single stuck-at fault for which no test has been generated.

3.19 Untestable Fault

A single stuck-at fault for which no test exists.

3.20 Unused Fault

A single stuck-at fault for which no test exists because the propagation path is floating.

4. PROCEDURE

The typical procedure flow for fault simulation and fault grading is shown in Figure 1.

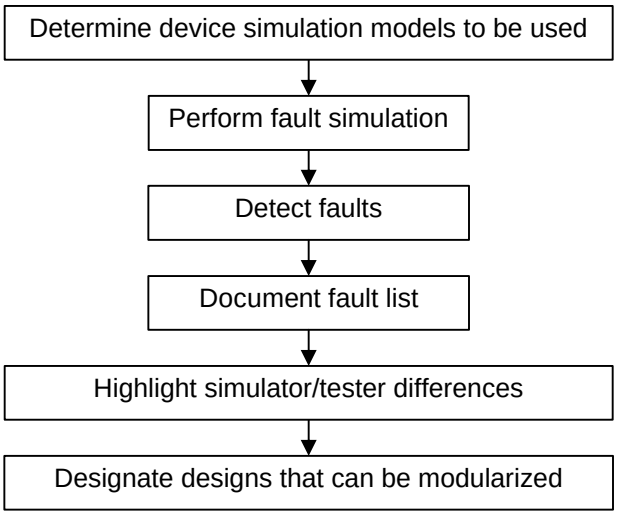


Figure 1: Typical Fault Simulation and Fault Grading Procedure Flow

4.1 Device Simulation

Simulation is an imitative process used to study relationships between parameters that interact in an Integrated Circuit. The simulator must support at least zero (0), one (1) and unknown (U or X) logic states. In addition, the simulator must support appropriate "strengths" to enable correct modeling of logic based upon the target technology and design practices.

Simulation employs models that are replica accurate enough to imitate the behavior of the circuit. Integrated circuits can be described at several levels of abstraction (see Figure 2):

a. Behavioral Model: The integrated circuit is described in terms of the algorithm that it performs.

b. Functional Model: The integrated circuit is described in terms of the flow of data and control signals within and between the functional blocks. These blocks are made of latches, registers, and elements of similar level of complexity.

c. Logical Model: The integrated circuit is described in terms of an interconnection of switching elements (gates and flip-flops) and is also referred to as gate or structural model.

d. Switch-Level Model: The integrated circuit is described in terms of the logical behavior of a metal oxide semiconductor circuit. A switch-level model consists of nodes connected by transistors, also referred to as transistor model.

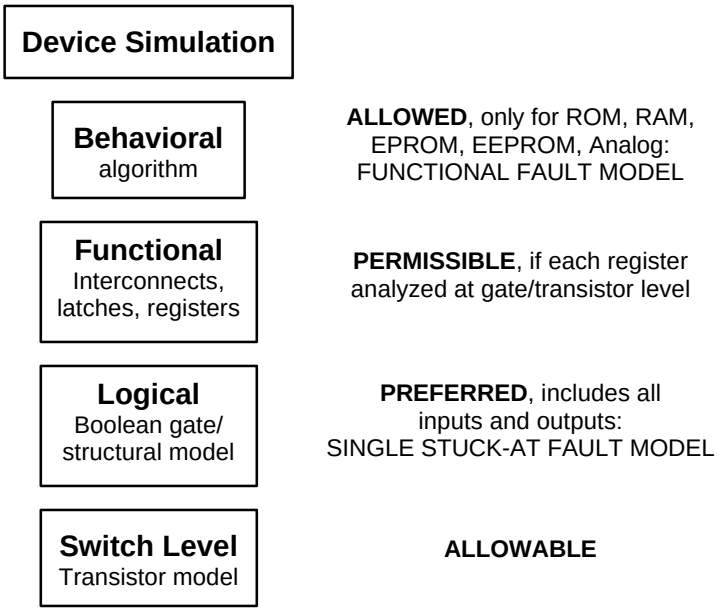


Figure 2: Device Simulation for Integrated Circuits

4.1.1 Simulation Model

A simulation model of the fault free device shall be constructed. Modeling of the device shall be at the Boolean gate level (Logical Model) and include all inputs and outputs. Modeling at the transistor level is allowable. Modeling at the register level is permissible if each register model is analyzed at the internal Boolean gate or transistor level for stuck-at-one (SA1) and stuck-at-zero test coverage with the test sequence applied to the external register nodes.

4.1.2 Simulation Database

The database used for simulation shall include all gates internal to the device, including memory portions, analog sections, and high impedance buffers to the input/output pins. Behavioral models will only be allowed to model the functionality of RAMs, ROMs, EPROMs, EEPROMs, and analog sections of design. Behavioral models on other modules may be used as long as the module under consideration for fault grading is modeled at the gate level.

4.2 Fault Models

Figures 3 and 4 illustrate the types of faults and fault models that are relevant to this document.

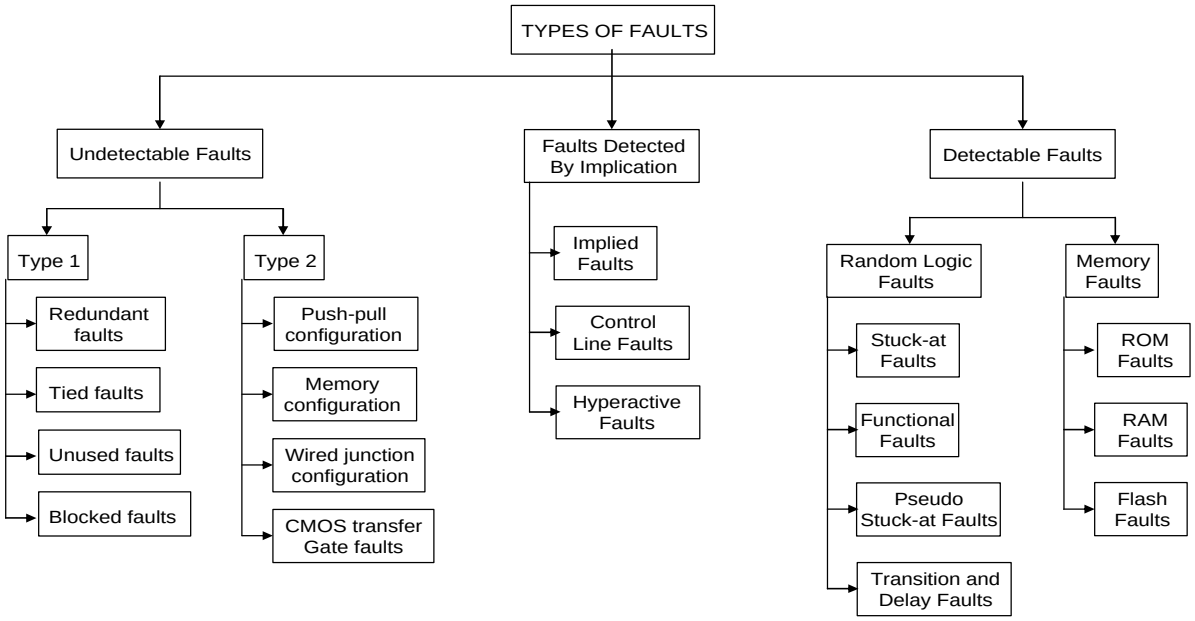


Figure 3: Fault Types

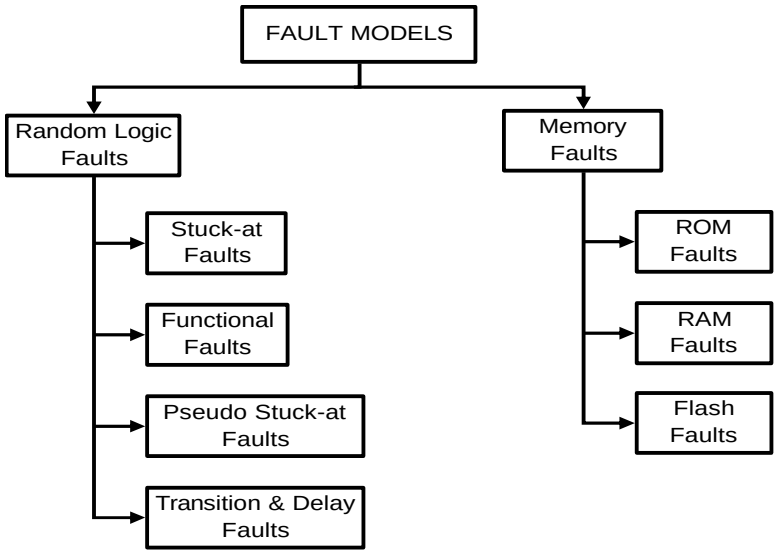


Figure 4: Fault Models

4.2.1 Detectable Fault Simulation for Random Logic

Fault simulation is used to measure the effectiveness of a defined ordered set of input test vectors to detect a specified set of modeled faults in a device under test. For each considered fault location, a fault is injected and a circuit simulation is executed to determine whether observed values differ from the behavior of a good circuit. The fault models and coverage metrics for random logic that are relevant to this document are discussed next.

4.2.1.1 Stuck-at Fault Model

A fault is defined as a single, stuck-at-one (SA1) or stuck-at-zero (SA0) condition (see Section 4.7 for more information). A fault model is constructed by injecting a fault at time zero (steady-state) into the fault-free simulation. An input stuck-at fault is assumed isolated from any fan-out branch that emanates from the output driving the input under consideration (see Figure 5).

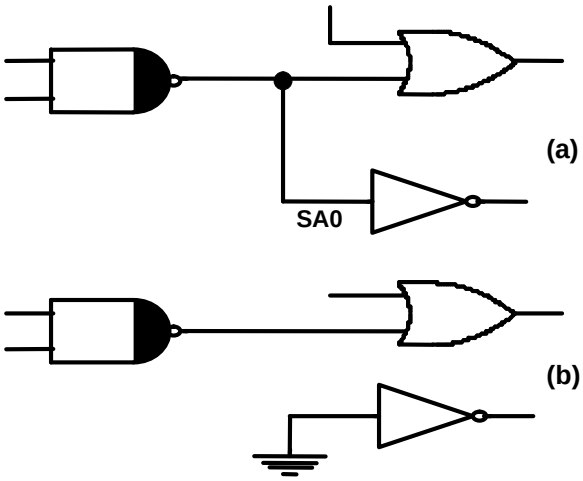


Figure 5: (a) Fan-out Logic (b) Same Logic with SA0

Two fault models shall be constructed for each gate input and each gate output, one simulating each fault type (i.e., SA1 and SA0). Each of these fault models must be tested for fault detection.

A stuck-at fault requires the following for detection:

a. Fault is excited (i.e., the node is placed into the logic state opposite to the fault state)

b. Fault effect is propagated at least to one device pin (i.e., primary output) or scan flip-flop data input

c. Primary output (i.e., scan chain output) is observed on the tester for the expected logic value

4.2.1.2 Functional Fault Model

Functional fault models are used to model defects at a level of abstraction that is much higher than in the single stuck-at fault model. Such models should be used only for designs using a behavioral model.

4.2.1.3 IDDQ Pseudo Stuck-at Fault Model

The pseudo stuck-at fault model is the most common fault model for IDDQ testing. It has the benefit of providing high defect coverage without requiring accurate transistor-level simulation models. Defects that may be missed by tests based on the traditional stuck-at fault model may be detected with IDDQ tests based on the pseudo stuck-at model. These defects include some adjacent bridging defects and CMOS transistor stuck-on conditions.

The difference between the stuck-at and pseudo stuck-at fault model is best described by comparing the fault detection requirements of each fault model.

4.2.1.3.1 The traditional stuck-at fault requires the following for detection:

a. Fault is excited (i.e., node is put into a logic state that is opposite to the fault state)

b. Fault effect is propagated to the device pin (i.e., primary output)

c. Primary output is strobed (i.e., repetitive measurements) by the tester for an expected logic value

4.2.1.3.2 An IDDQ pseudo stuck-at fault requires the following for detection:

a. Fault is excited (same as stuck-at fault)

b. Fault effect is ONLY propagated to the gate or cell output

c. IDDQ current measurement is done

4.2.1.4 Transition Delay Fault Model

Transition delay (TD) faults model large delay defects at the inputs or outputs of gates. The TD fault model is similar to the stuck-at fault model, but instead of a fault being described as stuck-at-one (SA1) or stuck-at-zero (SA0), a fault is described as being slow-to-fall or slow-to-rise. The TD fault models a gate input or output that is defective because its value is slow to change.

A transition delay fault requires the following for detection:

a. Fault is excited by launching a low-to-high or high-to-low transition on the node

b. Fault effect must be captured into a sequential element within a period of time that matches the functional at-speed timing

c. Fault effect is propagated to the device pin (primary output)

d. Primary output is strobed (i.e., repetitive measurements) by the tester for an expected logic value

Note: Items b, c, and d above can be replaced by one step in cases where the node combinationally connects to a primary output rather than a sequential element. In this case, the fault effect must reach a primary output that is strobed (i.e., repetitive measurements) within a period of time that matches the functional at-speed timing.

4.2.2 Detectable Fault Simulation for Memory

Memory simulation covers items such as coupling faults which are covered by most prominent march tests. Typically, memory built-in self test (BIST) covers a number of different memory faults which are hard to quantify or categorize.

4.2.2.1 ROM Faults

Read-only-memory is considered fully verified when all locations are read and faults occurring during read out may be propagated to an observable output.

4.2.2.2 RAM Faults

Random-access-memory is considered fully verified when each location can be detected in a stuck-at-one (SA1) or stuck-at-zero (SA0) condition and the address decode circuitry has been sufficiently exercised with faults propagated to an observable output. The test method should be documented and presented to the user upon request.

4.2.2.3 Additional RAM and ROM Tests

It should be realized that additional tests for RAM and ROM elements are generally required to detect topological and parametric faults.

4.2.2.4 Previously Graded Designs

Previously graded designs with only ROM code changes do not require a re-grading of the entire device as long as the ROM code, in either design, is not utilized in test grading other portions of the circuitry.

4.3 Fault Detection

4.3.1 Initial Condition

At the start of fault simulation, the state of every logic line and all components containing memory must be unknown (U or X). Any other initial condition, including explicit initialization of any line or memory element to zero (0) or one (1) must be justified and documented. If the same initialization is done in every instance of a specific model, then it is sufficient to document the initialization once. However, it must be stated that all instances of the model were affected.

4.3.2 Test Sequence

The device test sequence shall be introduced into the single stuck-at fault model and the propagation of signals simulated.

4.3.3 Detection Criteria

A fault is detected when a logical difference of values (i.e., between a 0 and a 1) at a device output exists between the fault-free model and the fault model. This difference is the result of the induced stuck-at condition.

4.4 Fault List

A fault list that refers to the set of all the modeled faults in a circuit must be generated in a deterministic approach. Statistical sampling of modeled faults is not permissible.

4.5 Documentation of Simulator / Tester Differences

Any differences in format or timing of the test vector sequence, between that used by the fault simulator and that applied by the tester, shall be documented in the fault simulation report.

4.6 Modularized Designs

Designs that may be modularized and tested independently of each other may be test graded separately and may not need to be redone for each design variation, as long as the test pattern for each module is always the graded pattern and each input and output is available when faults are scored.

4.7 Types of Stuck-at Faults

Figure 6 summarizes the type of stuck-at faults described and used in this test method.

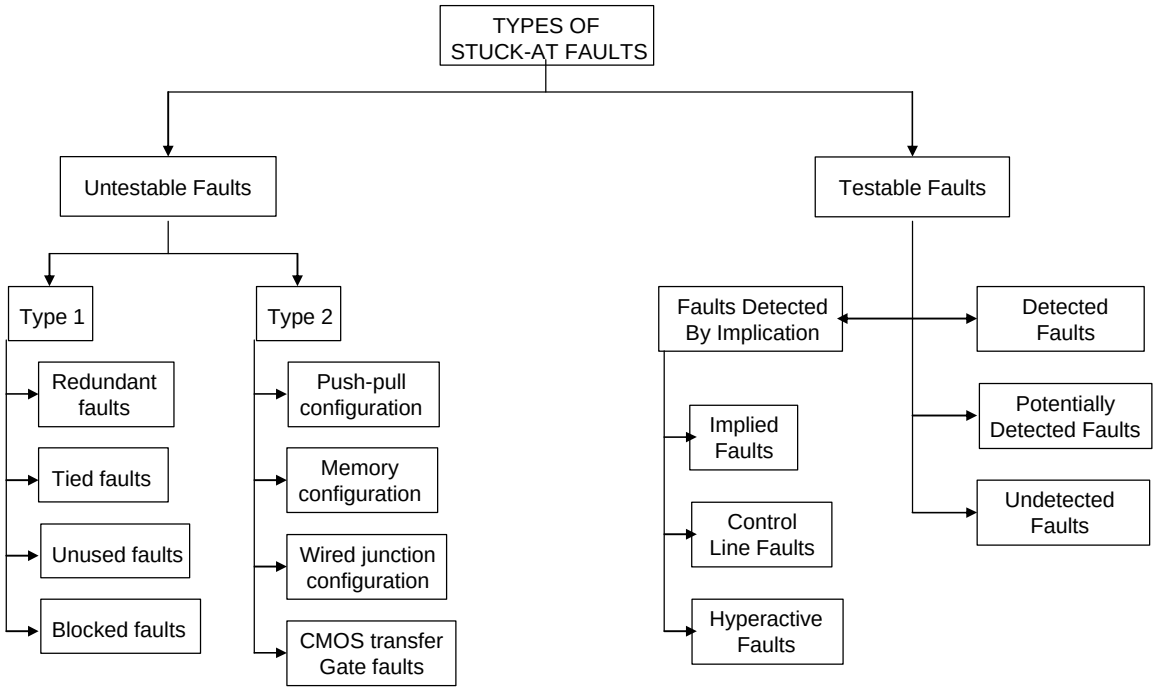


Figure 6: Types of Stuck-at Faults

4.7.1 Untestable Faults

Untestable faults are those faults that exist in the model and the actual circuit but cannot be verified by propagation to an observable output. Untestable gate input and output faults may exist in logic circuits and are generally caused by redundancies and unobtainable internal logic states. Untestable faults fall into two main categories: 1) TYPE1 faults that are completely untestable at both the logic and circuit level, and 2) TYPE2 faults that may be testable at the circuit level but are untestable at the logic level. Logic-level effects are defined as Boolean-type effects in which a faulty logic zero (0) occurs instead of an expected logic one (1), or vice versa. Fault simulators can typically only grade faults with logic-level effects. Circuit-level (also called parametric) effects are defined as only causing modifications of the voltage, current, or signal propagation time. The signal modifications may not be significant enough to have a Boolean effect. Fault simulators are typically incapable of simulating TYPE2 effects, and thus, will never be able to indicate detection of them regardless of the patterns.

TYPE1 untestable faults represent defects that would not have detrimental effects on the device. Thus, faults proven to be TYPE1 may be removed from fault list at an early state of test grading. TYPE2 faults, on the other hand, may have circuit-level effects. However, these effects cannot be seen in fault simulators. Thus, TYPE2 faults cannot be graded by fault simulators and must be covered by other methods.

TYPE1 and TYPE2 faults fall into several subcategories that are described herein.

4.7.1.1 TYPE1 Untestable Faults

This section describes the primary categories of TYPE1 untestable faults. TYPE1 faults are completely untestable at both the logic and circuit level. These faults represent defects that would not have any detrimental effects on the device. Other kinds of TYPE1 faults, beyond those defined herein, may exist.

4.7.1.1.1 Redundant Logic

If a design contains logical redundancy, the faults associated with the redundant logic are truly untestable and can be deleted from the fault list. However, it is desirable that the design be modified to remove the redundancy if it is unintentional. This section is concerned with true logic redundancy and not the type of redundancy that is often employed to improve circuit performance (see Figure 7).

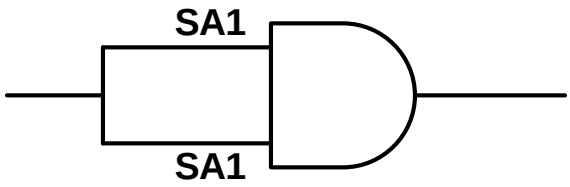


Figure 7: SA1 on either one of the inputs is untestable redundant

4.7.1.1.2 Tied Logic

Tied faults include faults on gates where the point of the fault is tied to a value identical to the fault stuck value. The tied circuitry could be due to tied signals. Examples are stuck-at-one (SA1) on the power supply and stuck-at-zero (SA0) on ground (see Figure 8). Also, the tied circuitry could be due to an AND gate with complementary inputs or an exclusive-OR gate with common inputs.

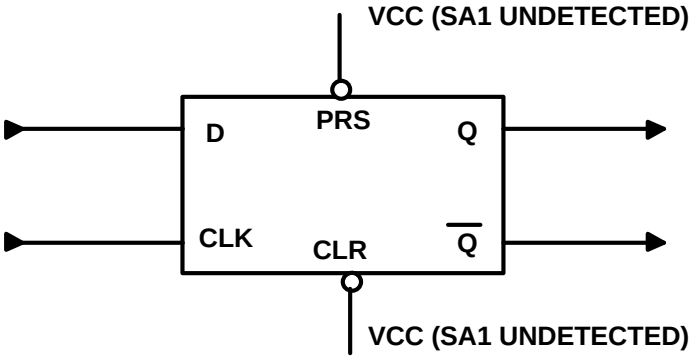


Figure 8: SA1 on both inputs is untestable tied

4.7.1.1.3 Unused Logic

Circuitry that has no connectivity to an externally observable point is considered unused logic. An output of a flip-flop that is unconnected to any other circuit will have a stuck-at-one (SA1) and stuck-at-zero (SA0) fault which will both be considered unused (see Figure 9).

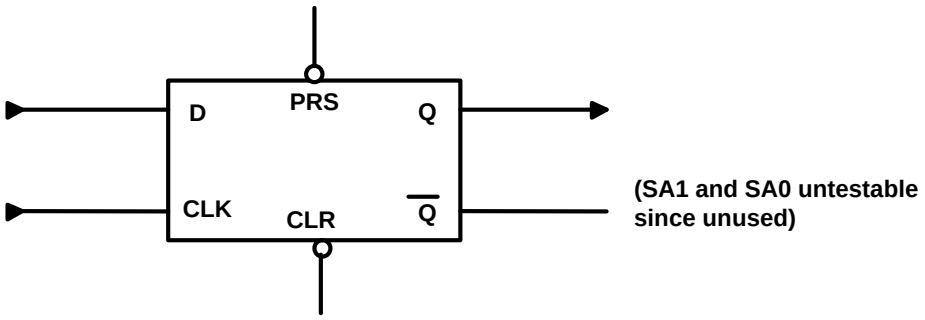


Figure 9: SA1 and SA0 on Q-bar are untestable unused

4.7.1.1.4 Blocked Logic

Blocked faults include faults on circuitry for which tied logic blocks all paths to an observable point.

4.7.1.2 TYPE2 Undetectable Faults

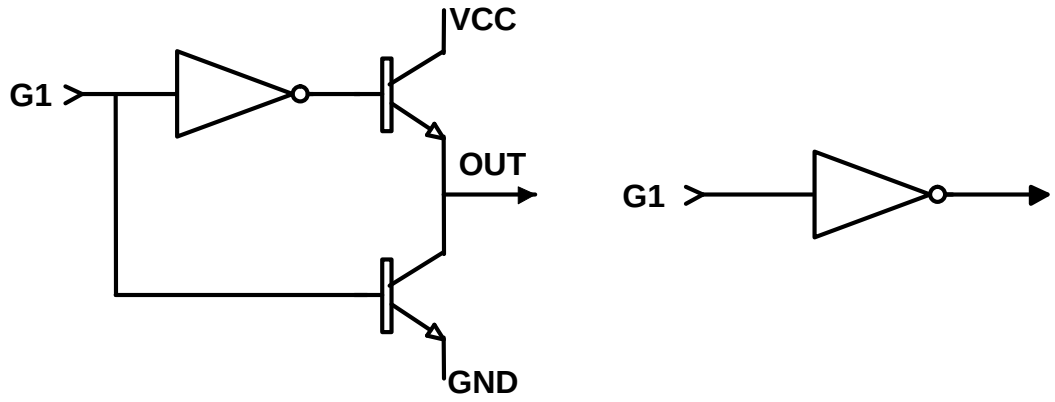
TYPE2 undetectable faults have no logical effects but may have parametric or circuit-level effects. The inability of present fault simulators to simulate circuit (in addition to logic) fault effects makes it impossible to detect TYPE2 faults through simulation. Other methods are required to ensure coverage.

Also, it is important to note that TYPE2 faults often are only visible in custom logic in which many pieces of the design may be modeled at the transistor level. Newer designs composed of standard cells and based on a synthesized flow usually do not have TYPE2 faults. This is due to the modeling of the standard cells. The circuit level details of the standard cells are not modeled for the logic and fault simulations. Consequently, the only kind of undetectable faults in standard cell designs will be the TYPE1. Although TYPE2 faults may not be visible in standard cell designs, defects may still occur which only have parametric and circuit-level effects similar to TYPE2 fault effects. Methods to cover such defects are presented in later sections.

This section describes many examples of TYPE2 undetectable faults. Other kinds of undetectable TYPE2 faults may exist that are not described in this section.

4.7.1.2.1 Push-Pull Configurations

Undetectable faults in a push-pull configuration may be subtracted from the total number of faults or the configuration may be modeled as a buffer or inverter (see Figure 10).



(a) Push-Pull (b) Equivalent Model

Figure 10: Push-Pull Configuration

4.7.1.2.2 Memory Configuration

Undetectable faults in a memory configuration may be subtracted from the total number of faults or the memory configuration may be modeled as a functional memory element (see Figure 11).

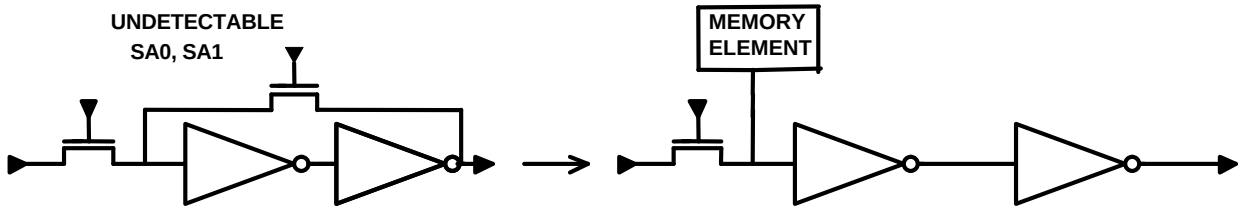


Figure 11: Undetectable Faults in a Memory Configuration

4.7.1.2.3 Wired Junction Configuration

Undetectable faults in a wired junction configuration with no dominance may be removed from the total number of faults (see Figure 12). However, in many wired junction configurations all stuck-at faults are detectable and are required to be counted in the total number of faults graded.

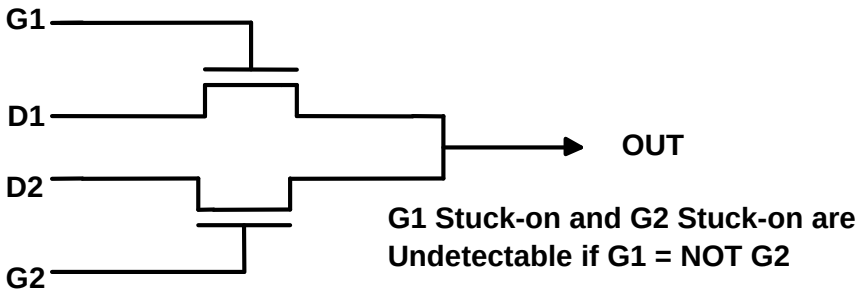


Figure 12: Undetectable Fault in a Wired Junction Configuration

4.7.1.2.4 CMOS Transfer Gate Faults

An example of a CMOS transfer gate undetectable fault is illustrated in Figure 13 below. A SA0 fault on the gate of N transistor has no effect on logic state transmission through the transfer gate despite the N transistor remaining off. A simulation of the parametric effects of the threshold drop is beyond the capabilities of present fault simulators.

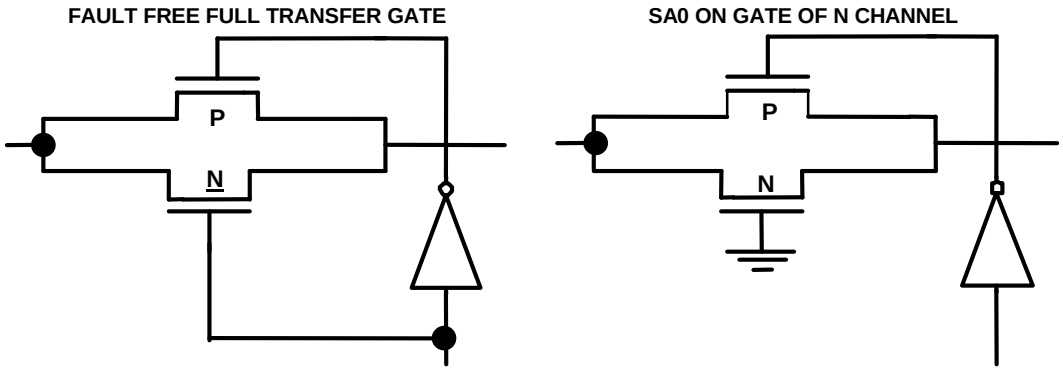


Figure 13: CMOS Transfer Gate Undetectable Fault - Circuit Fault Effect

It should be noted that such faults are not necessarily undetectable in the device under test. Otherwise, at least part of the logic involving these faults could have been deleted from the actual design. However, such faults can not be detected by observing stuck at fault effects, and parametric testing (e.g., speed DC drive, etc.) is necessary for covering such faults. It is, therefore, acceptable to exclude such faults from the fault list because the test coverage definition pertains to stuck-at faults only.

4.7.2 Testable Faults

4.7.2.1 Faults Detected by Implication

This section describes many examples of faults detected by implication. Other kinds of faults detected by implication may exist that are not described in this section.

4.7.2.1.1 Implied Faults

If a fault exists on an internal node that cannot be initialized to a known value, but the fault collapses into another dominant fault that can be detected, then the fault is considered detected by implication (see Figure 14). Implied faults must be analyzed on a case-by-case basis.

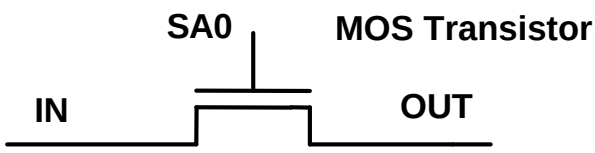


Figure 14: Control SA0 will be detected if (OUT) SA0 and SA1 faults are detected when no other sources fan-in to signal OUT (Implied Fault Detection)

4.7.2.1.2 Control Line Faults

In the absence of initialization circuitry, a fault on a control line will cause the output of the component to be at U or X. An example is SA0 or SA1 fault on the CLK input of a D flip-flop that has no set and reset inputs (see Figure 15). Therefore, the control line fault can at best be potentially detected. However, if both SA0 and SA1 faults on the data input are detected, the two faults on the control line can be counted as detected if the U or X at the output is assumed to be a permanent zero (0) or a permanent one (1). This assumption is allowed and such faults must be documented as having been detected by implication.

Figure 15: CLK faults will be detected if SA0 and SA1 faults are detected on D or Q

4.7.2.1.3 "Hyperactive" Faults

A fault may generate a lot of simulation activity, such as an oscillation, causing the program to stop due to the overwhelming demand on a fault simulator. This fault is also known as a hypertrophic fault.

4.7.2.2 Detected Faults

A single stuck-at fault for which a test has been generated.

4.7.2.3 Potentially Detected Faults

A single stuck-at fault for which a test has been generated but is not 100% effective in detecting this fault.

4.7.2.4 Undetected Faults

A single stuck-at fault for which no test has been generated.

5. TEST COVERAGE MEASUREMENTS

The following measurements are mainly focused on the Stuck-at fault model and very likely not applicable for other models, such as I_{DDQ}.

5.1 Fault collapsing

To facilitate fault simulation, the concept of fault equivalence and dominance is allowed to be used. Fault equivalence and dominance allow us to combine many faults into a single set and a single test vector can detect these faults. The process of reducing the total number of possible faults into a minimal number of necessary faults is called fault collapsing. For example, faults in a series of buffers can be combined (collapsed) into one set of faults.

5.2 Potentially Detected Faults

A modeled fault is considered potentially detected if during application of the test vectors, a primary output value in the fault-free logic model is a zero (0) or one (1) at a specific simulation time, but goes to U or X in the corresponding faulted logic model at the same simulation time. A fault that is potentially detected at least 10 times can be considered a detected fault. This is based upon the assumption that the U or X value will be opposite of the fault-free logic model value at least once if it occurs 10 times or more during the application of test vectors. Most fault simulators allow the user to set a threshold value for this purpose, which must be set to at least 10. An alternative method is to simply count all potentially detected faults and consider 50% of that amount as detected faults.

5.3 Test Coverage

5.3.1 Percent of Faults Detected

The percent of faults detected, or test grade, is equal to the total number of faults detected divided by the total number of possible faults minus undetectable faults.

$$\text{Test Coverage (\% Detected)} = \frac{\text{[Total \# Detected Faults]}}{\text{[Total \# Possible]} - \text{[\# Undetectable]}} \times 100\%$$

where:

Total # Possible = 2 X [Total # of Gate Inputs + Total # of Gate Outputs]

A rule-of-thumb for selecting an appropriate calculation method is that the logical (gate) fault model is used for digital devices and blocks using automated test pattern generation (ATPG).

5.3.2 TYPE2 Faults

Since the effects of TYPE2 faults cannot be quantified by standard simulation tools, indirect methods should be used to ensure coverage of these faults. TYPE2 faults often can be considered a delay fault in digital circuits. For analog circuits, TYPE2 faults can affect the parametric specifications.

For scan designs, scan patterns targeting delay faults must be included. Functional patterns may be used to supplement scan for testing delay faults. For non-scan designs, delay faults must be covered solely by functional patterns.

The capture cycle of the scan patterns intended to target delay faults should execute at the rated frequency of the device and may also be used to cover stuck-at faults. For functional patterns, tests should be run at the rated frequency of the device. The delay fault test description and coverage should be reported per Section 5.3.1.

For the testing of analog circuits, which are outside the scope of this document but may be specified elsewhere, functional tests, parametric tests, or other types of tests (such as Built-In-Self-Test) that verify the specifications are required. Test descriptions should be reported per Section 7.

5.3.3 Coverage Reporting

Interim fault coverage reporting may be based on a collapsed fault list. The final reported test coverage however, shall be in terms of the total number of faults in the fault list, not the collapsed fault list. The fault coverage percentage reported must be divided into models (e.g., stuck-at, transition delay, IDDQ) and device segments (e.g., analog, digital logic, memory) as appropriate.

5.3.4 Algorithm Derived Test Vectors

If an established test algorithm is used to derive test vectors for parts of designs for which a behavioral model is used, the established test coverage must be reported. References and other relevant material must be documented in support of the effectiveness of the algorithm used. If an established test algorithm is customized or a new test algorithm is developed, its effectiveness must be proved and the test coverage (so established) should be reported. If a behavioral model contains sub-block(s) that are modeled at structural level (i.e., decoding logic associated with RAM partitions), justification must be provided in the fault simulation report as to how the stuck-at faults in the embedded structural logic are covered by the test algorithm that was used.

5.3.5 Automatic Test Pattern Generation (ATPG) / Scan Testing

In scan-based designs, scan test patterns may be used instead of functional patterns to provide the specified stuck-at test coverage requirements. If the specified stuck-at test coverage cannot be met with only scan patterns, functional patterns may be used to supplement the scan patterns.

Additional tests are generally required to detect delay faults. In scan-based designs, scan patterns targeting delay faults are required. These scan patterns may be supplemented by functional patterns. To detect delay faults, the functional tests should be run at the rated frequency. For scan tests targeted for delay faults, the capture cycles should execute at the rated frequency. The delay fault test description and coverage should be reported per Section 5.3.1.

6. ACCEPTANCE CRITERIA

6.1 Statistical Sampling

Statistical sampling of modeled faults is not permissible.

6.2 Qualification Test Requirements

Devices submitted for qualification and approval must be tested using a vector set with stuck-at fault coverage greater than or equal to the percentages shown below for parts of designs for which a logic model is used. If the test coverage requirements are not met, an explanation of all undetected faults along with plans for improvement must be submitted. If applicable, the performance of I_{DDQ} is encouraged unless the supplier provides an explanation for why I_{DDQ} cannot or should not be performed. To reach the level of coverage required in this document, the device must be designed to accommodate I_{DDQ} testing. The following are the production test coverage requirements for different segments of the device and depending on the existence and level of I_{DDQ} testing.

6.2.1 Analog Circuits or Analog Circuit Blocks of Mixed Mode Circuits

100% specification coverage is required.

6.2.2 Digital Circuits or Digital Circuit Blocks of Mixed Mode Circuits

The stuck-at fault coverage of the production test set used for all parts delivered for production must be greater than or equal to **98%** test coverage.

6.2.3 Digital Circuits or Digital Circuit Blocks with I_{DDQ} or I_{SSQ}

If acceptable I_{DDQ} or I_{SSQ} testing (per the Appendix) is included in the production test set, the stuck-at fault coverage of the production test set used for all parts delivered for production must be greater than or equal to **97%** test coverage.

6.2.4 Transition Delay Fault Coverage for Digital Circuits or Digital Circuit Blocks Utilizing Scan Designs

The addition of transition delay fault coverage is desirable to raise the overall test coverage of the device. A reasonable target is **80%** test coverage.

6.2.5 Digital Circuits or Digital Circuit Blocks using Pseudo Stuck-at I_{DDQ} Fault Coverage

The addition of pseudo stuck-at I_{DDQ} fault coverage is desirable to raise the overall test coverage of the device. A reasonable target is **70%** test coverage.

6.3 Theoretical Field Reject Rate

The theoretical reject rate of the test coverage can be calculated using its percentage along with the functional yield according to the models referenced in Agrawal and Williams-Brown. These model estimates are theoretical worst-case estimates and only account for stuck-at faults. Inclusion of other types of faults greatly complicates these calculations, so care must be exercised in using these results.

6.4 Test Sequence Alterations

Following the acceptance of test grading, no test deletions will be allowed without performing a new test grade on the entire test sequence for the affected circuit block(s). However, additional tests may be added. Also, for every revision of the design, the acceptable test coverage level must be re-established because some of the previously successful tests may get invalidated due to design modifications. Should the supplier have sufficient means to preclude the need for a new test grade following a change, the results must be provided to the User prior to production deliveries.

6.5 User Audits

The User reserves the right to audit the results of fault and test grading.

6.6 Failure to Meet Production Fault Coverage

If the production test coverage requirement cannot be met, the supplier must submit a full report to the User for approval, explaining the reasons that the requirement cannot be met.

7. DOCUMENTATION

The documentation delivered must include the following in the specified order:

a. Statement of test coverage that includes percent test coverage, the number of faults detected, the total number of faults, number of detectable faults, and the undetectable faults for each device segment and model used.

b. Breakdown of fault simulation results by logic blocks per a top level description showing:

   1. Equitably distributed fault coverage.

   2. Where behaviorally modeled logic used.

c. Description of logic and ATPG/fault simulation tool used.

d. Potential fault detection threshold used (minimum of 10).

e. Description of tests targeting delay faults and analog parametrics for TYPE2 coverage.

f. Details of fault coverage (cite references) by Built-In Self-Test methodologies, if any used.

g. Detail any differences in format or timing of the test vector sequence, between that used by the fault simulator and that applied by the tester.

h. I_{DDQ} pseudo stuck-at coverage for the selected I_{DDQ} vector subset, if used, and the distribution of measured I_{DDQ} values and the upper acceptance limit.

Appendix

IDDQ Testing

If the supplier elects to use IDDQ testing, the following applies:

1. A team consisting of design, product, and reliability engineers shall review the test vectors and select a set of IDDQ vectors that provide a minimum pseudo stuck-at coverage specified in Section 6.2.5, based upon their best engineering assessment. In all cases, this shall not be less than ten (10) test vectors, unless it can be demonstrated that a greater test coverage as specified in Section 6.2.5 can be achieved with fewer test vectors.

2. The test program will be modified to pause on the select IDDQ test vectors and record the total current from the positive power supplies. The current from any analog supply must not be included in this measurement. Although the IC clock has been paused during these IDDQ test vectors, integrity of the IC internal data and its outputs shall not be compromised. IDDQ measurements must also allow sufficient time for the IC to settle into its quiescent operating mode.

3. For production material and production-intent material for qualification, the failure criteria shall be calculated from matrix material used for electrical characterization, or the first three qualification lots. For each matrix cell (excluding cells with intentional Leff variation), a minimum of twelve (12) functional devices shall be selected and their IDDQ values recorded. This data will then be statistically analyzed for its mean and standard deviation, assuming a normal distribution. If the data suggests otherwise, the supplier should illustrate the method of analysis. The upper IDDQ acceptance criteria shall be the mean plus seven standard deviations or less. This acceptance criteria may not apply for deep sub-micron technology processes where intrinsic leakage is much higher than defect-induced currents. For this case, acceptable methods include but are not limited to Ringo-based IDDQ limits, neighborhood screening, and Δ-IDDQ.

4. At any time during the product's lifetime, the supplier may submit distribution information, reliability data, and failure analyses supporting a change to the IDDQ test limit.

5. IDDQ testing may be more effective if employed after a voltage stress test. Any such test combination (e.g., voltage stress and IDDQ) must have the IDDQ test following the voltage stress.

6. IDDQ tests shall be run at the maximum operating power supply voltage(s) defined in the device electrical specification.

7. All parts shipped for manufacturing shall have IDDQ testing included in the wafer test or final test programs. A single temperature is acceptable.

8. The use of a pre/post stress delta IDDQ or ISSQ methodology is desirable.

Automotive Electronics Council
Component Technical Committee

Revision History

Rev #	Date of change	Brief summary listing affected sections
-	Sept. 6, 1996	Initial Release.
A	July 18, 2003	Corrected formatting errors. Added new captions to all Figures.
B	Sept. 18, 2007	Complete Revision. Revised Acknowledgment section and added Notice Statement. Revised sections 1, 4.1, 4.1.1, 4.1.2, 4.2.1, 4.2.1.1, 4.2.1.2, 4.3.1 to 4.3.3, 4.4 to 4.6, 4.7.1, 4.7.1.1.1, 4.7.1.2.1 to 4.7.1.2.3, 4.7.2.1.1, 4.7.2.1.2, 5.1 to 5.3, 5.3.1 to 5.3.5, 6.2, 6.2.2, 6.2.3, 6.4 to 6.6, 7, and Appendix; and Figures 5, and 8 to 15. Added new sections 2, 3, 3.1 to 3.20, 4, 4.2, 4.2.1.3, 4.2.1.3.1, 4.2.1.3.2, 4.2.1.4, 4.2.2, 4.2.2.1 to 4.2.2.4, 4.7, 4.7.1.1, 4.7.1.1.2 to 4.7.1.1.4, 4.7.1.2, 4.7.1.2.4, 4.7.2, 4.7.2.1, 4.7.2.1.3, 4.7.2.2 to 4.7.2.4, 5, 6.2.1, 6.2.4, 6.2.5, and 6.3; and Figures 1 to 4, 6, and 7. Deleted old sections 3.1.8, 3.2, 3.2.1, 3.2.2, 3.3, 3.3.1 to 3.3.4, 4.2.3, and 4.3; and old Figures 2, and 8 to 10.