

IEEE Standard for Analog Defect Modeling and Coverage

IEEE Computer Society

Developed by the
Test Technology Committee

IEEE Std 2427™-2025

STANDARDS

IEEE Standard for Analog Defect Modeling and Coverage

Developed by the

Test Technology Committee
of the
IEEE Computer Society

Approved 10 September 2025

IEEE SA Standards Board

Abstract: A defect coverage accounting method based on simulation models for defects observed within integrated circuits (ICs) is defined in this standard. The portion of a defect universe, comprising thousands or millions of reasonably likely defects, that is detected or “covered” by tests of analog and mixed-signal circuits depends on many factors, which this standard considers, such as detectability, process variations, defect characteristics, and redundancy. The contents of a defect coverage summary are specified and dozens of commonly used terms are clearly defined to aid communication about the quality of tested ICs.

Keywords: AMS test, analog/mixed-signal test, analog test coverage, defect coverage, defective parts per million, design for test, DFT, DPPM, IEEE 2427™

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2026 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.
Published 9 January 2026. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-2776-0 STD28383
Print: ISBN 979-8-8557-2777-7 STDPD28383

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all IEEE standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. “Artificial intelligence” refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/about/contact/>.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

At the time this standard was completed, the Analog Defect Coverage Working Group had the following members and past members:

Anthony Coyette, *Chair*
Geoffrey Coram, *Vice Chair*
Devanathan Varadarajan, *Secretary*
Anne Meixner, *Editor*
Stephen Sunter, *Past Chair*
Salem Abdennadher, *Past Vice Chair*

Inga Abel
Andreas Baenisch
Amit Bajaj
Lakshmanan Balasubramanian
Paul Berndt
Mayukh Bhattacharya
Ken Butler
Kevin Cameron
Tapan Chakraborty
Degang Chen
Bjoern Ebert
Heiko Ehrenberg
Chris Everson
Michele Fava
Ken Ferguson
Karthik Ganesan
Renaud Gillon

Jo Gunnes
Amit Gupta
Koutaro Hachiya
Dieter Haerle
Ian Harrison
Lars Hedrich
Artur Hovsepyan
Atul Jain
Andrei Karavaev
Kazuyuki Maruo
Ashok Mathur
Yasuyuki Miyahara
Zain Navabi
Kurt Neugebauer
Pedro Ovalle
Rubin Parekhji
Vidhya Prasad Ravindran
Jeff Rearick

Mike Ricchetti
Christophe Sabtier
Peter Sarson
Keno Sato
Art Schaldenbrand
Gregor Schatzberger
Ahmed Sokar
Fei Su
Tejan Thakore
Bernard Ting
Louis Ungar
Ronny Vanhooren
Hans Martin von Staudt
John Wang
David Wu
Vladimir Zivkovic
Mark Zwolinski

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Salem Abdennadher
Inga Abel
Lakshmanan Balasubramanian
Paul Berndt
Mayukh Bhattacharya
Kenneth Butler
Kevin Cameron
Degang Chen
Geoffrey Coram
Anthony Coyette
Heiko Ehrenberg

Christian Everson
Renaud Gillon
Ian Harrison
Kazumi Hatayama
Werner Hoelzl
Dieter Joos
Krzysztof Jurga
Andrey Karavaev
Adam Ley
Ian McIntosh
Anne Meixner
Jaafar Mejri

Kurt Neugebauer
Jim O'Reilly
Rubin Parekhji
Jeff Rearick
Saghir Shaikh
Stephen Sunter
Louis Ungar
Devanathan Varadarajan
Hans Martin von Staudt
Vladimir Zivkovic
Mark Zwolinski

When the IEEE SA Standards Board approved this standard on 10 September 2025, it had the following membership:

Lei Wang, *Chair*
Jon Walter Rosdahl, *Vice Chair*
David J. Law, *Past Chair*
Alpesh Shah, *Secretary*

Edward Au
Ted Burse
Xiaofeng (Alfred) Chen
Doug Edwards
Nehad El-Sherif
J. Travis Griffith
Deborah R. Hagar

Guido R. Hiertz
Ronald W. Hotchkiss
Tyler L. Jaynes
Thomas Koshy
Howard Li
Xiaohui Liu
Kevin W. Lu
Hiroshi Mano

Daleep C. Mohla
Annette D. Reilly
Robby Robson
Daniel Sabin
F. Keith Waters
Sha Wei
Luyang (Eric) Zhang

Introduction

This introduction is not part of IEEE Std 2427-2025, IEEE Standard for Analog Defect Modeling and Coverage.

The development of this standard began a month after the first occurrence of the “Emerging Test Strategies” session in the 2014 European Test Symposium. Four informal presentations described the state-of-the-art for analog test and design-for-test (DfT), and why it appeared to lag that of digital DfT by more than 30 years. It was observed that industry’s focus for analog circuitry in integrated circuits (ICs) was primarily on reducing test cost, a little on DfT automation, and minimally on measuring test coverage. It was proposed that the order of priorities needed to be reversed: first focus on measuring test coverage, then developing test/DfT automation, and lastly on reducing test cost. It was also clear that this was too big a problem for any company to solve alone.

After that ETS, Peter Sarson proposed to the presenters and a few of the most vocal session attendees that they meet periodically to try to close this gap by creating a standard. Prior experience developing the IEEE Std 1149.1™, the IEEE Std 1149.4™, and the IEEE Std 1687™, each of which required many years, led to the conclusion that this small group should first develop a framework for a standard and then propose a Working Group to complete the task. Monthly meetings quickly became weekly meetings, and then in 2017 a formal IEEE Study Group was formed, with over 70 participants from dozens of companies and universities. The group proposed what became in 2018 the P1687.2 Analog Test Access Working Group and the P2427 Analog Defect Coverage Working Group. Many people continued to be members of both Working Groups (which met on alternate weeks) and new members joined one or both groups.

Acknowledgments

The working group acknowledges JEDEC for its permission to use the definition of “latent defect” from JEDEC JESD 88F:2018 [B18].

The working group acknowledges ISO for permission to use the definitions of “fault” and “error” from ISO 26262-1:2018 [B17] in Annex A.

Contents

1. Overview	12
1.1 Scope	12
1.2 Purpose	12
1.3 Context	13
1.4 Organization of the standard.....	14
1.5 Word usage.....	14
2. Definitions, acronyms, and abbreviations	15
2.1 Definitions	15
2.2 Acronyms and abbreviations	22
3. Defect classification and defect coverage reporting	24
3.1 Defect universe.....	25
3.2 Defect models and defect injection.....	37
3.3 Defect coverage.....	49
3.4 Detectable defect coverage	55
3.5 Defect weighting.....	57
3.6 Redundancy and defect tolerance	60
3.7 Defect Coverage Summary.....	62
Annex A (informative) Review of definitions of defect versus fault in the literature	65
A.1 General definitions.....	65
A.2 Hard, catastrophic, gross faults/defects	66
A.3 Soft, parametric, deviation faults/defects.....	67
Annex B (informative) Benchmark circuits and defect lists.....	68
Annex C (informative) Defect simulation failure.....	69
C.1 Common failure scenarios	69
C.2 Broad classification.....	71
Annex D (informative) MOS transistor open defect models	73
D.1 Introduction	73
D.2 Defect models	73
D.3 Test circuit setup.....	74
D.4 DC operating points.....	75
D.5 MOSFET models	75
D.6 Assessing gate leakage.....	76
D.7 Simulator controls.....	77
D.8 Simulation results and waveforms	78
Annex E (informative) Example coverage reports	84
E.1 Example Defect Universe Details for LNA-based amplifier.....	85
E.2 Example Defect Coverage Summary for LNA-based amplifier.....	87
E.3 Example Defect Coverage Details for LNA-based amplifier.....	89
Annex F (informative) Relationship between IEEE Std 2427-2025 and IEEE P1687.2	92
Annex G (informative) Added value of this standard, and expected users.....	93
Annex H (informative) Comparing analog defects with faults described in other standards	95

H.1 Differences in definitions.....	95
H.2 Reporting coverage metrics	97
Annex I (informative) Typical defect terminology and categories.....	99
I.1 Hard (catastrophic) and parametric (soft) defects	99
I.2 Defect categories	101
I.3 Defect models.....	102
I.4 Latent defects	102
Annex J (informative) Estimating escaping parts in defective parts per million (DPPM)	103
J.1 State of the art	103
J.2 Defect level estimation for a fault/defect coverage with a single fault model	104
J.3 Defect level estimation with multiple fault models.....	104
J.4 DPPM estimation—from digital faults to analog defects.....	105
Annex K (normative) List of IEEE Std 2427™-2025 recommendations and permissions, paraphrased ...	106
Annex L (informative) Bibliography.....	108

IEEE Standard for Analog Defect Modeling and Coverage

1. Overview

1.1 Scope

This standard defines a defect coverage accounting method based on simulation models for defects observed within integrated circuits (ICs). The portion of all possible defects that are detected or “covered” by tests of analog and mixed-signal circuits depends, in practice, on many factors (detectability, defect characteristics, detection threshold margin, measurement resolution, operating point, test patterns, etc.). In defining the defect coverage report process, the standard considers most of these factors.

This standard focuses on defects in analog functions. In this context, “defect” is an unintended physical change in a circuit, and an “analog function” means a function that has input, internal, or output signals with meaningful values in a defined continuous range. The function has at least one parametric performance that is sufficiently non-deterministic that its test has upper and/or lower limits (the limits can be real numbers or quantized digital equivalents).

This standard considers redundancy, since most analog circuits have redundancy and defect tolerance, intentional or not. This standard does not consider combinations of variations that could result in a circuit failing to meet all of its specifications; that is the subject of Monte Carlo simulations during design for yield and multi-parameter analog defect modeling.

The defects considered here are applicable to purely digital circuits too, though typically a simplified fault model (stuck-at, for instance) is utilized for digital functions. Thus, this standard assumes the topic of stuck-at fault coverage accounting is addressed for digital circuits by IEEE Std 1804™, IEEE Standard for Fault Accounting and Coverage Reporting (FACR) for Digital Modules [B16].⁶

1.2 Purpose

The primary purpose of this standard is to allow people to communicate information about defect coverage in a way that allows assessment of test and circuit quality as well as prediction of important metrics [simulation time, design for test (DfT) circuit area, test time, test escapes, etc.]. Given a circuit description and a set of tests, this standard defines how to enumerate the universe of defects, determine which defects are detected by the tests, and report the coverage results.

⁶ The numbers in brackets correspond to those of the bibliography in Annex L.

By defining analog test coverage, this standard is useful for several other purposes.

First, this standard guides efficient simulation of defects to ensure that the defect models achieve a desirable trade-off between cost-effective simulation times and accurate modeling of behavior seen in real circuits.

Second, this standard facilitates estimation of test escape rates (historically measured in defective parts per million [DPPM]) to facilitate trade-offs between cost of test, time to market, and quality.

Third, this standard facilitates improvements in DfT and test generation methods. Many DfT and test techniques, including built-in self-test (BIST), have been developed. Despite their cost advantages, they are not used in practice because their impact on defect coverage is questioned. This is due to the lack of a well-understood and silicon-corroborated analog defect model. This standard facilitates automation and quality improvements that enables reliable comparisons of DfT and test techniques.

1.3 Context

Unlike for digital circuits, the industry has lacked a standardized way to specify the coverage of tests for analog and mixed-signal circuits. This omission has led to, at best, the use of incompatible ad hoc definitions or, at worst, no quantitative consideration of analog defect coverage at all. The anticipated benefits of using this standard include the following:

- Definition of a defect universe for a given analog circuit (schematic or layout netlist)
 - A list of defect types and their typical root causes
 - A practical and realistic model for each defect type
 - Equivalent defects that facilitate defect collapsing and simulating fewer defects
- Standard test coverage metrics for analog/mixed-signal circuits
 - A clear definition of defect detection criteria
 - A clear definition of defect coverage whereby different tools give comparable estimates for a given circuit and test
- Requirements for analog defect simulators; standard ways to automatically:
 - Inject defects
 - Measure the ability of the tests to detect defects in circuit elements
 - List undetected defects
 - Report coverage of a circuit by a set of tests in an objective, traceable way
- A feedback mechanism (i.e., coverage measurement) that allows:
 - Creation of higher coverage tests, for a defined set of defects
 - Reduction of test time without reducing test coverage
- A more deterministic way to estimate DPPM for IC devices containing analog circuitry
 - Based on a measure of defect coverage, accounting for defect clustering and defect likelihoods

- A way to report in internal reports and external papers, the coverage achieved by new test and DfT techniques, to allow more objective comparison to other approaches
- A basis for evaluating defect-oriented analog automatic test pattern generation (ATPG)
- Analog defect injection and simulation definitions/rules, which could be extended to multi-parameter variation defects in the future

Future benefits needed and related to analog defect modeling/coverage that are not presently addressed in the standard include the following:

- Replacing specification-based tests with defect-oriented tests
- Reduced analog defect simulation time, for a given set of defects and tests

1.4 Organization of the standard

After the overview, the next section provides a set of definitions, abbreviations, acronyms, and special terms. This is followed by an extensive set of specifications, which include for each subclause a description, rules, recommendations, and permissions. Finally, annexes provide informative references that provide general background information, technical explanations related to several aspects of circuit simulation and how the standard could be implemented, etc. It also includes a list of references cited in the standard. One normative annex provides the list of all permissions and recommendations with a paraphrased description that is required in one of the three deliverables.

1.5 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{7,8}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

⁷ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁸ The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Definitions, acronyms, and abbreviations

2.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁹

Some terms have been informally defined in the literature. Formal definitions for these are provided for clarity in the context of defect coverage and modeling.

This standard's Working Group members concluded that a "defect" is a physical entity, whereas a "fault" is an unacceptable performance of a primitive circuit. The standard relies on definitions for "defect" and not "fault." Nevertheless, "fault"-related definitions are still included in this standard for two reasons. First, as Annex A shows, published papers demonstrate an inconsistency in their differences between definitions for "defect" and "fault." The IC test community's haphazard interchange of "defect" and "fault" would likely continue without an established standard for these definitions, which is herewith rectified. Second, this standard addresses analog test coverage from purely a defect viewpoint. A future standard could address analog fault coverage and use the definitions herein.

For the purposes of clear identification of deliverables, the following terms are capitalized: Defect Coverage Details, Defect Coverage Summary, and Defect Universe Details.

ac coupling defect: *See: capacitive connection defect.*

application circuitry: Any circuit elements whose operation could affect a specified device performance.

capacitive connection defect: A capacitance between two nodes that is much larger than the original parasitic capacitance between them. Unlike other open defects, this defect creates an unintended ac coupling path between the nodes.

Example: 30 fF coupling between adjacent metal interconnects caused by poor optical proximity correction (OPC).

circuit: A group of connected circuit elements designed to perform a function.

circuit element: A fundamental electrical structure defined in the process design kit (PDK) for an integrated circuit (IC) process.

Examples: resistor, capacitor, inductor, diode, transistor.

circuit module: A group of connected circuit elements and/or primitive circuits and/or lower-level circuit modules that has specified performance at its primary ports.

NOTE—Defect coverage is more easily measured and reported for accessible circuit modules, that is, modules that have a fully defined test access.¹⁰

Examples: An amplifier comprising only transistors and resistors; a filter comprising an amplifier plus resistors that attenuate the input signal; a flash digital-to-analog converter (DAC) comprising a series of 32 resistors, selector transistors, logic gates, and an output filter.

⁹*IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

¹⁰Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

circuit under test: A circuit undergoing test via simulation or during a manufacturing test insertion or during an in-system self-test.

collapse: To group defects that have the same effect on circuit performance and select only one of the defects to simulate.

NOTE 1— Two defects are structurally equivalent if they produce the same circuit topology when they are applied non-simultaneously; this would be formally provable.

NOTE 2— Two defects can be considered behaviorally equivalent if they produce the same circuit behavior when they are applied non-simultaneously; this might not be formally provable.

Examples: Grouping an open drain defect and an open source defect for a transistor to be just an open drain defect—for a logic circuit this represents a behavioral equivalence. Grouping a short defect across each of three diodes in parallel and selecting only one short to simulate—this represents a structural equivalence.

connection: An ideally lossless path between circuit elements, ports, and/or terminals; the path could be electrical, optical, etc.

defect: An unexpected physical change in a circuit element or connection between circuit elements, that is not within fabrication specifications for the circuit element or connection, regardless of whether the circuit using that defective circuit element or connection is within its specifications or not.

NOTE 1— Process parameter values of a design-intent circuit element are expected to deviate from their typical values within a PDK-specified acceptable range.

NOTE 2— For brevity, in the context of a circuit simulator, defect refers to an instance of a defect model.

NOTE 3— “Fabrication specifications” refer to the manufacturing rules and procedures as prescribed by the manufacturer. This is referred to in the process design kit (PDK).

NOTE 4— When there are fabrication specifications for the maximum mismatch between adjacent circuit elements, mismatch in excess of those specifications is a defect. When there are no such specifications, the excessive mismatch is not a defect but causes a fault if the circuit no longer meets its specified performance.

defect coverage: The weighted percentage of defects detected by tests applied to the circuit’s ports as a fraction of all defects in the circuit’s defect universe.

Defect Coverage Details: The list of defects simulated and their detection status.

NOTE—Rule 3.3f) specifies two options for documenting Defect Coverage Details in the Defect Coverage Summary.

Defect Coverage Summary: A file or report that states the defect coverage, circuit netlists, tests, process models and defect models used, etc.

NOTE—Subclause 3.7 defines the complete requirements for the Defect Coverage Summary.

defect model: An equivalent circuit for a defect in the context of a circuit simulator.

NOTE—See 3.2 for this standard’s descriptions for defect model associated with the various defects defined.

Examples: A short defect modeled as a 300- Ω resistor; excessive second-order temperature coefficient of a resistor; gate oxide thickness of a metal-oxide-semiconductor (MOS) transistor significantly less than expected.

defect universe: All reasonably likely defects in a circuit that can be algorithmically defined for one or more defect types, e.g., short, open, parametric, excluding those in any unused circuitry.

NOTE—Subclause 3.1 specifies how the defect universe is to be enumerated according to this standard.

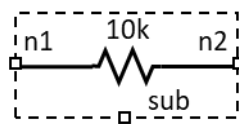
Defect Universe Details: A required set of descriptions of the defects in the defect universe for which a Defect Coverage Summary is provided.

NOTE—Subclause 3.1 specifies contents of the Defect Universe Details.

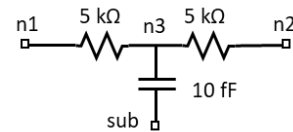
design-intent circuit element: A circuit element that does not include parasitic circuit elements or other non-idealities of its physical implementation.

Example: A schematic symbol showing a design-intent resistor that is ideally a single 10-k Ω resistance: in the simulated circuit netlist, two 5-k Ω resistors and a parasitic capacitance to the substrate are instantiated as a subcircuit model of the 10-k Ω resistor.

Defects might be injected in the single 10-k Ω resistor or in each of the resistors in the subcircuit model.



A schematic symbol showing design-intent resistor



Resulting simulated circuit schematic

Figure 1—Design-intent circuit element and its representation in a schematic

detectable defect: A defect for which a conceivable test exists that would fail in the presence of that defect.

NOTE—This is the default designation for all defects.

detected defect: A defect for which a test fails in the presence of that defect.

detectable defect coverage: The weighted percentage of all detectable defects in a circuit's defect universe detected by tests applied to a circuit's ports as a fraction of all detectable defects in the circuit's defect universe.

detection: The event when either 1) a defect causes a value measured at a circuit's ports to fail a test that passes defect-free and fault-free circuits subjected to within-specification process parameter variations, or 2) a circuit element of the circuit exceeds safe operating area (SOA) limits by so much that immediate destruction of a large area of the chip occurs.

NOTE—SOA violations are, in general, not valid detection methods in simulation, since most simulators do not model device damage due to operation outside the safe operating area.

device: A monolithic integrated circuit die (packaged or not) or a multi-chip package that contains multiple dies.

fail: The result of a test when the tested behavior is not within the test limits.

Example: An output slew rate that is less than the lower test limit but possibly not less than the lower specification limit.

failure: A device behavior whose function or performance ceases to meet specifications.

Example: A power consumption that is greater than the datasheet upper specification limit.

fault: An unexpected change in a specified performance of a primitive circuit or circuit module that is not within the circuit's specification limits, regardless of whether the performance of circuitry using the faulty primitive circuit or faulty circuit module is within its specifications or not.

NOTE 1— Deviations from the typical values of parameters of a primitive circuit are expected, if they are within a specified acceptable range.

NOTE 2— The change could be physical or electrical and need not be permanent. For example, a single event upset (SEU) that changes a flip-flop state is an unexpected change that is not within specification limits and is usually an unacceptable change that could disappear in the next clock cycle.

Example: An amplifier with gain less than the lower specification limit.

fault model: A general scheme for modeling faults in the context of a circuit or behavioral simulator.

Example: The stuck-at model for logic.

fault universe: All reasonably likely faults that can be algorithmically defined, plus all defects in the defect universe that are not within the primitive circuits that have faults.

hard defect: A defect that permanently changes a circuit's topology: an open in a connection, or a short between nodes.

NOTE 1— Often referred to as a catastrophic defect.

NOTE 2— In this context, "permanently" means the defect is present at least until the end of the test being simulated.

Examples: A stringer between adjacent polysilicon interconnect lines; a short between source and drain of a metal-oxide-semiconductor (MOS) transistor; an open in a metal conductor.

hard fault: A fault for which an output port of a primitive circuit incorrectly responds to stimuli applied to one or more of its input ports.

NOTE—Also called a catastrophic fault.

Examples: A NOT AND gate (NAND) whose output does not respond to one of its inputs, regardless of the state of its other inputs; an oscillator that cannot be disabled by its power-down input; an op-amp whose output oscillates when at a temperature within the op-amp's specified temperature range.

intermittent defect: A defect that manifests itself inconsistently during random time intervals.

NOTE 1— When activated, it can be described as a short, open, or parametric defect.

NOTE 2— Intermittency could occur due to unstable connections or stress.

latent defect: A physical defect inherent in the process architecture, design, or layout, or created during manufacturing (wafer fabrication or assembly) that is manifested after some period of operation (from JEDEC JESD 88F:2018).¹¹

leakage defect: A leakage path, typically across a thin oxide or PN junction, with such a high resistance that it could be modeled as a resistor or a current source.

marginal defect: An unintended, and unwanted but insignificant manufacturing flaw that is within fabrication specifications for the circuit element or connection but could become a defect if voltage differences, current flow, process window shift, thermal effects, or aging cause additional change to the circuit due to inadequate design for manufacturability (DfM)/design for reliability (DfR).

NOTE—When activated, it can be described as a short, open, or parametric defect.

merge: To replace multiple circuit elements with a single circuit element to reduce simulation time.

NOTE—Most circuit simulators automatically merge circuit elements internally to reduce simulation time.

Example 1: Replacing four instances of identical transistors in parallel by a single instance that has a multiplication factor of $M=4$.

Example 2: Replacing five instances of non-identical resistors in series (that does not have intermediate nodes connecting to other circuit elements) by a single resistor having a resistance equal to the sum of the five resistances.

multi-element independent parametric defect: A defect that consists of circuit elements in close proximity in which at least two of the circuit elements each have a different parametric defect.

Example: A group of transistors, one of which has V_{th} too high, one has L too short, and one has W too narrow.

multi-element parametric defect: A defect that consists of circuit elements in close proximity that have the same parametric defect.

Example: Four parallel transistors, all having V_{th} that is too high; a resistor and capacitor, both having excessive etching that causes excessive resistance and insufficient capacitance.

multi-element parametric mismatch defect: A defect that consists of non-defective circuit elements in close proximity that have process parameter values that do not match within process specifications.

Example: A pair of non-defective resistors have identical design layouts, but the ratio of their actual resistances is not within process specifications.

node: A point at which one or more terminals of circuit elements are connected.

open defect: A defect that breaks a connection between two (or more) terminals of one or more circuit elements of the defect-free netlist.

parametric defect: A defect that permanently changes one or more process parameters of a circuit element.

NOTE—In this context, “permanently” means the defect is present at least until the end of the simulation.

¹¹ Reproduced with permission of JEDEC, Copyright ©2018.

Examples of process parameters: oxide thickness (t_{ox}), low-field mobility (μ_0), voltage threshold (V_{th0}).

parametric fault: A fault implemented as a change in a primitive circuit's or circuit module's netlist that only parametrically affects its performance.

NOTE—Figure 2 illustrates a parametric fault and associated fault model for an integrator circuit.

Examples: A too-high corner frequency of a filter; insufficient slew rate of a driver; excessive jitter in a phase-locked loop (PLL) output signal.

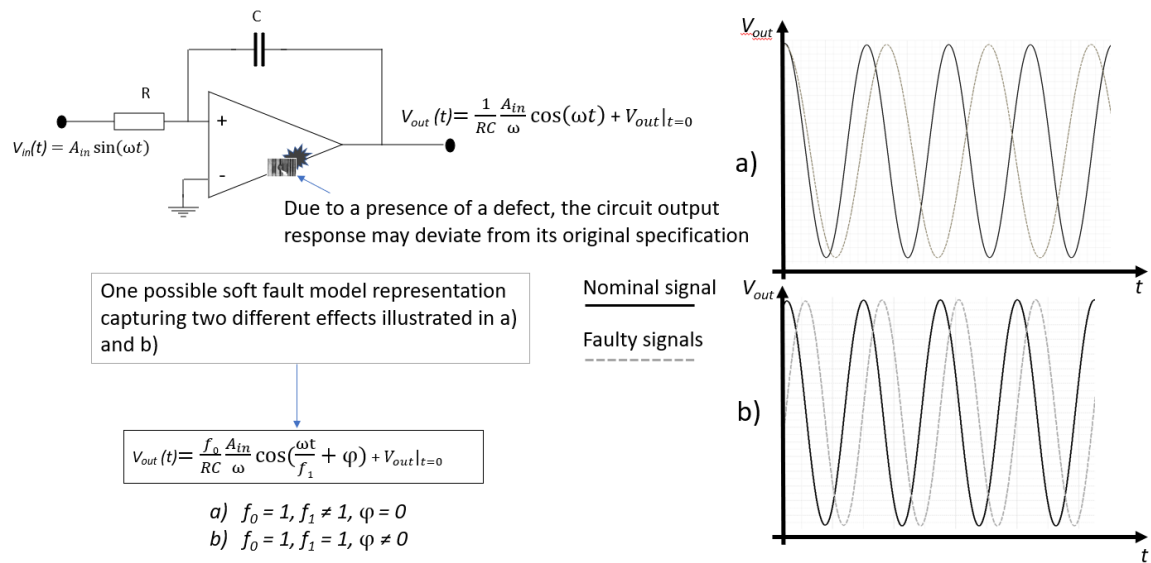


Figure 2—Integrator circuit with defined behavior model and a parametric fault model

port: A node of a circuit that is externally accessible to convey a signal or power (via voltage, current, light, magnetic field, movement, heat, etc.). Performance at a port, or combination of ports, can be characterized to derive test limits.

potentially undetectable defect: A defect that is identified, and supported by a justification, as likely to not be detected by any conceivable test, for the defect model used.

NOTE 1— The defect could become detectable if a suitable test method is developed. Examples of such test methods include adding DfT and adding adaptive test limits to an existing test.

NOTE 2— A change in a defect model (in either a value or circuit topology) used could cause some defects categorized as potentially undetectable defects to become detectable defects. Conversely, a change in a defect model could cause some detectable defects to become potentially undetectable defects.

Examples: When N is large enough, an open in one of N parallel identical circuit elements, or a short in one of N series identical circuit elements.

primary port: A port that has a specified performance.

primitive circuit: A circuit that has specified performance(s) and does not contain any subcircuits that have specified performance(s).

Examples: Transmission gate; amplifier; comparator; bandgap voltage reference; latch; flip-flop; combinational logic gate.

process corner: A realistic combination of process parameter values that results in maximum performance deviations from typical, for which the process manufacturer or the designer deems a design is intended to meet all the design's specifications with acceptable yield.

Example: One corner is usually Slow N-channel metal-oxide-semiconductor (NMOS) & Fast P-channel metal-oxide-semiconductor (PMOS) ("SF").

process design kit: A set of files used to model the basic circuit elements of a fabrication process for the tools used to design and simulate an integrated circuit.

random defect: A defect whose occurrence and location in a device are random and statistically independent of defects occurring elsewhere in the device or occurring in the same location in other devices.

Examples: In integrated circuits (ICs), these defects are typically caused by particles on the wafer's surface during processing, or inherent fluctuations in dopant or etching.

resistive connection defect: A defect that can be typically modeled as an open defect with a resistance less than 1 M Ω . The resistance typically affects only a signal in a connection between a low impedance driver and a low impedance load.

resistive short defect: A defect that can be typically modeled as a short with resistance greater than 10 k Ω but less than 1 M Ω . The resistance typically affects only high impedance signals.

safe operating area: The voltage and current conditions over which a circuit element can be expected to operate without self-damage.

short defect: A defect that creates an additional conductive electrical path between two (or more) nodes in the defect-free netlist.

soft fault: *See:* parametric fault.

specified performance: A functional performance characteristic that has pre-determined values at significant times, and/or specifications with upper and/or lower limits.

SPICE-level: A representation of circuit functions using a netlist of basic circuit elements (resistor, capacitor, inductor, diode, metal-oxide-semiconductor (MOS) transistor, bipolar transistor, etc.) and their ideal models or models based on measurements of their individual physical implementations.

structurally undetectable defect: A defect for which it can be demonstrated that, due to the structure of the circuit in which the defect is located, no test can be developed to detect the defect.

NOTE—The understanding of structurally undetectable defect is only for single defects. If combined with another defect, a structurally undetectable defect could become a detectable defect.

Example: Short across a resistor that ties an input to a rail.

systematic defect: A defect whose occurrence or location in a device is statistically correlated to defects occurring elsewhere in the device or occurring in the same location in other devices.

Examples: In integrated circuits (ICs), these defects are typically caused by imperfections in a manufacturing apparatus, a processing step, or a design characteristic, leading to localized shifts in optical focus, exposure, mask overlay, or etch rates.

terminal: One of a circuit element's connecting points.

test: A procedure to assess the quality of a device by applying one or more environmental conditions and electrical stimuli and observing the response in terms of defined limits for a defect-free device within acceptable process variations, without causing damage to a device that has no defect or latent defect.

NOTE 1— For a circuit netlist, the stimuli and expected response are applied and observed at the circuit's ports.

NOTE 2— Environmental conditions are applied to the entire device, such as temperature, pressure, humidity, or physical stress.

NOTE 3— Limits can be defined post-processing; for instance, using statistical outlier analysis such as part average testing (PAT).

transient fault: A fault whose effect is not permanent.

trim: To permanently adjust each device's tunable circuitry during the manufacturing test flow to reduce the impact of acceptable process variations on its performance metrics.

NOTE—The term originates from physically trimming a circuit by mechanical action or with a laser, but it is now usually accomplished by programming non-volatile memory (NVM).

undetectable defect: A defect that cannot be detected by any conceivable test.

undetected defect: A defect for a considered process corner for which none of the tests fail in simulation.

unused circuitry: Any circuit elements whose functionality (or lack thereof) will not affect the intended application of the device under any specified operations.

NOTE—Circuit elements could be considered as unused circuitry even if they receive stimulus in the intended application, or if a defect in them could affect a different application.

2.2 Acronyms and abbreviations

AMS	analog/mixed-signal
ATPG	automatic test pattern generation
BIST	built-in self-test
CL	confidence level
CMOS	complementary metal-oxide semiconductor
CMP	chemical mechanical polishing
CUT	circuit under test
DAC	digital-to-analog converter

DF_PDK	deviation factor relative to PDK process corner
DfM	design for manufacturability
DfR	design for reliability
DfT/DFT	design for test, or design for testability
DPPM	defective parts per million
DSPF	detailed standard parasitic format
DUT	device under test
EDA	electronic design automation
EOS	electrical over stress
ESD	electro-static discharge
HCI	hot carrier injection
HDL	hardware description language
IC	integrated circuit
IP	intellectual property
KCL	Kirchhoff's current law
KVL	Kirchhoff's voltage law
LNA	low-noise amplifier
LWRS	likelihood-weighted random sampling
MOS	metal-oxide semiconductor
MOSFET	metal-oxide semiconductor field-effect transistor
MPF	multi-point faults
NAND	NOT AND gate
NBTI	negative bias temperature instability
NMOS	N-channel metal-oxide semiconductor
NVM	non-volatile memory
OPC	optical proximity correction
PDK	process design kit
PMOS	P-channel metal-oxide semiconductor

PVT	process, voltage, temperature
SEU	single event upset
SOA	safe operating area
SoC	system-on-chip
SPEF	standard parasitic exchange format
SPICE	simulation program with integrated circuit emphasis
TDDDB	time-dependent dielectric breakdown
VCVS	voltage-controlled voltage source

3. Defect classification and defect coverage reporting

This standard provides instructions for measuring defect coverage using circuit simulations and reporting the coverage. Figure 3 illustrates a general flow for achieving this standard's deliverables.

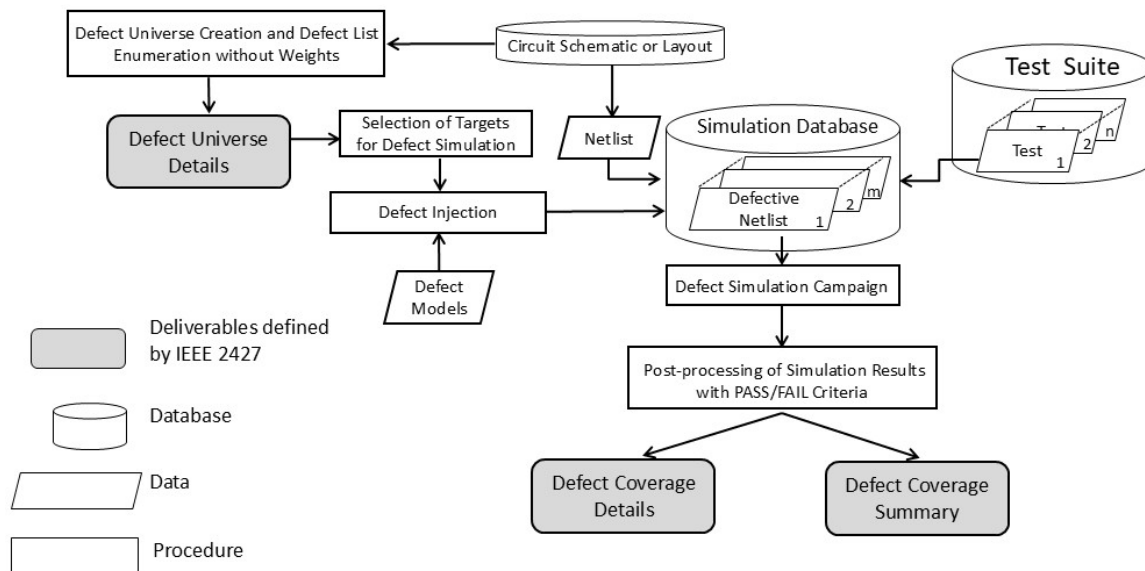


Figure 3— Deployment of IEEE 2427™ circuit defect simulation and defined deliverables

An analog or mixed-signal circuit typically consists of a hierarchy of subcircuits. For simulation, each subcircuit can have different model types or netlists to allow trade-offs between simulation speed and accuracy. The most common levels of abstraction for subcircuits consist of behavioral models, schematic netlists, and layout-extracted netlists. This standard only applies to transistor-level netlists.

Netlists generated by schematic capture software represent the designer's intent. Depending upon the process design kit (PDK), the resulting netlist might also include the parasitics associated with each design-intent resistor, capacitor, and transistor. In general, every node in a schematic corresponds to a node in the actual circuit layout.

Layout-extracted netlists are automatically extracted from a graphical layout that corresponds to the intended physical layout. These netlists include all *significant* parasitic capacitances, resistances, and

inductances. The design engineer sets the threshold for *significant* based upon the operating range of the circuit, including frequency, voltage, current, and other aspects. Also, extracted netlists can include other parasitic devices, for example, well diodes. The most commonly used standard formats include basic simulation program with integrated circuit emphasis (SPICE), detailed standard parasitic format (DSPF), and standard parasitic exchange format (SPEF).¹²

Similar to the single fault injection assumption in digital gate-level fault simulations, current industry practice in the mixed-signal domain is to inject a single defect per simulation. In addition to supporting the single-defect-per-simulation approach, this standard provides rules to support engineers who plan to simulate multiple independent defects in a single simulation.

3.1 Defect universe

This subclause describes the content of the defect universe. The defect universe can be defined for a circuit at two points in the design cycle, that is, based on a netlist extracted from a schematic or based on a netlist extracted from a layout or graphically derived from layout. This permits prospective users of the resulting defect coverage numbers to execute their engineering tasks; see Annex G for associated engineering roles.

The instructions in this subclause prescribe how the details of each defect are to be documented in a file or report called Defect Universe Details. This permits a level of detail that some users of the standard, for example, electronic design automation (EDA) tool developers and design automation engineers, find essential to their role in supporting accurate and time-efficient analog defect simulation. Note that not all users of the standard will have access to this level of detail; it might be too confidential.

These instructions cover circuit changes with respect to connectivity (topological changes) and to circuit elements' parameters. Specifically, defects in circuits fall into three main categories: missing connections, additional connections, and changes to parameters of existing circuit elements. Hard defects considered by the standard are “shorts” (additional connections) and “opens” (missing connections). An excessive change in a circuit element's parameter(s), such as a change in the resistance of an element, would be considered a parametric defect. A simple LNA-based circuit has been used in Figure 4, Figure 5, and Figure 9 to illustrate the various defect types, and Annex E includes the Defect Universe Details for this circuit.

The Defect Universe Details provide a means to communicate the nature of each defect, its location, and defect model used. Single defect insertion is the norm for most analog defect simulations, but this standard also supports the injection of multiple independent defects by providing instructions in Rule 3.1k).

Hard defects can be inserted in multi-fin, multi-channel, multi-gate, and multi-finger transistors the same way that hard defects are injected for a transistor instance that has only one gate, that is, in and between the terminals of the netlist instance, not within the transistor. Similarly, defects can be injected this way in transistors whose types are not explicitly mentioned in this standard.

During system-on-chip (SoC) integration of intellectual property (IP) blocks, especially third-party IP blocks, it is possible that some of the circuitry in an IP block would not be used in a specific SoC and could be considered unused circuitry from this standard's perspective. When defining the defect universe, it is necessary to distinguish application circuitry from unused circuitry without ignoring whether they share a power grid. If they share power rails VDD and VSS, for example, a short between power rails in the unused circuitry could influence the intended application. Therefore, such a defect belongs in the defect universe unless there is a topologically equivalent short within application circuitry and the short's weighting is increased. However, if other potential defects in this unused circuitry could affect the intended application by, for example, increasing power consumption or causing noise on VDD, then the circuitry belongs in the defect universe.

¹² SPEF is part of IEEE Std 1481™-1999 [B14] and IEEE Std 1481™-2009 [B15].

Recommendations for parametric defects are to have at least two defects per element type as determined by the engineer defining the defect universe. Recommendation 3.1n) suggests using an excessively high value and a low value of a circuit element parameter.

Merging and collapsing are different ways of reducing the number of defects explicitly listed in the Defect Universe Details without reducing the number of defect sites represented from a netlist. The reduction can simplify engineering analysis of undetected defects; it might improve the confidence interval when sampling defects. Also it can reduce the number of defects to simulate when not sampling defects. However, it does not reduce the total defect weight. Merging combines equivalent circuit element instances, and collapsing combines equivalent defects. The following example shows how collapsing reduces an original 30 defects in 10 elements to two Defect Universe Details entries, and merging reduces the number to two entries, or to three entries if one is added according to 3.2s). See Annex E for another example.

Example netlist before merging (in SPICE syntax) is as follows:

```
R0 n1 n2 10k
R1 n1 n2 10k
...
R9 n1 n2 10k
```

Defect Universe Details with no collapsing and no merging are as follows:

Serial#	Instance	Nodes	Value	Defect	Weight
Defect1	R0	n1 n2	10k	short	0.2
Defect2	R0	n1 n2	10k	open_n1	2.5
Defect3	R0	n1 n2	10k	open_n2	2.5
Defect4	R1	n1 n2	10k	short	0.2
Defect5	R1	n1 n2	10k	open_n1	2.5
Defect6	R1	n1 n2	10k	open_n2	2.5
...					
Defect28	R9	n1 n2	10k	short	0.2
Defect29	R9	n1 n2	10k	open_n1	2.5
Defect30	R9	n1 n2	10k	open_n2	2.5

Injecting a short replaces the netlist instance, for example, R0, with (in SPICE syntax):

```
R0      n1 n2      10k
Xdefect n1 n2      short_model
```

Injecting an open_n1 replaces the netlist instance, for example, R0, with (in SPICE syntax):

```
Xdefect n1 nopen open_model
R0      nopen n2 10k
```

Injecting an open_n2 replaces the netlist instance, for example, R0, with (in SPICE syntax):

```
R0      n1 nopen 10k
Xdefect n2 nopen open_model
```

Defect Universe Details with collapsing opens according to Recommendation 3.1u) are as follows:

Serial#	Instance	Nodes	Value	Defect	Weight
Defect1	R0	n1 n2	10k	short	0.2
Defect2	R0	n1 n2	10k	open	5 collapsed=2
Defect3	R1	n1 n2	10k	short	0.2
Defect4	R1	n1 n2	10k	open	5 collapsed=2
...					
Defect19	R9	n1 n2	10k	short	0.2

```
Defect20 R9          n1 n2 10k  open          5  collapsed=2
```

Injecting a short replaces the netlist instance, for example, R0, with (in SPICE syntax):

```
R0          n1 n2      10k
Xdefect    n1 n2      short_model
```

Injecting an open replaces the netlist instance, for example, R0, with (in SPICE syntax):

```
R0          n1 nopen 10k
Xdefect    n2 nopen open_model
```

Defect Universe Details with additional collapsing of opens according to Recommendation 3.1t) and shorts according to Recommendation 3.2p) are as follows:

```
Defect1  R0          n1 n2 10k  short          2  collapsed=10
Defect2  R0          n1 n2 10k  open           50  collapsed=20
```

Netlist after merging R0~R9 according to Permission 3.1x) is as follows:

```
R0_9 n1 n2 10k m=10
```

Defect Universe Details after merging according to Rule 3.1l) and Permission 3.1x), collapsing according to Recommendation 3.1t) and Recommendation 3.1u), and adding an open according to Permission 3.2s)

```
Defect1  R0_9          n1 n2 10k  short          2  merged=10
Defect2  R0_9          n1 n2 10k  1ofN_open    50  merged=10  collapsed=20
Defect3  R0_9          n1 n2 10k  open           1  merged=10
```

Injecting 1ofN_open replaces the netlist instance, for example, R0_9, with (in SPICE syntax):

```
R0_9a      n1 n2      10k m=9
R0_9b      n1 nopen 10k
Xdefect    n2 nopen open_model
```

Rules

- a) In the application circuitry netlist, each unique terminal of a design-intent circuit element shall be the site of an open defect.

Example: See Figure 4.

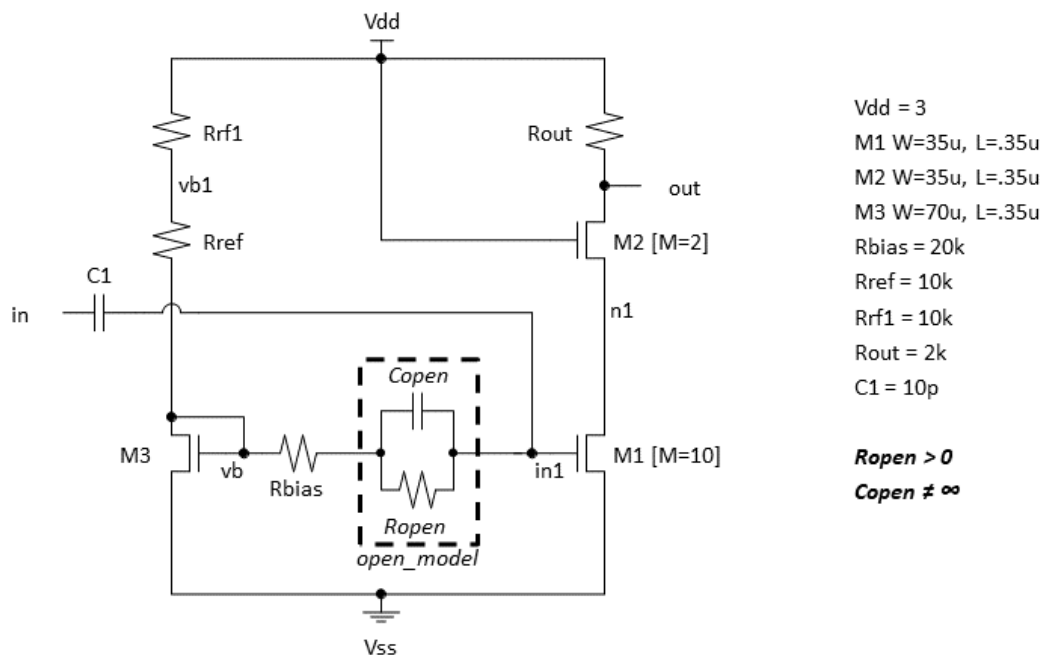


Figure 4—Example of an LNA with an open defect in Rbias

- b) In the application circuitry netlist, each pair of circuit nodes connected to terminals of a circuit element shall be the site of a short defect.

Example: See Figure 5.

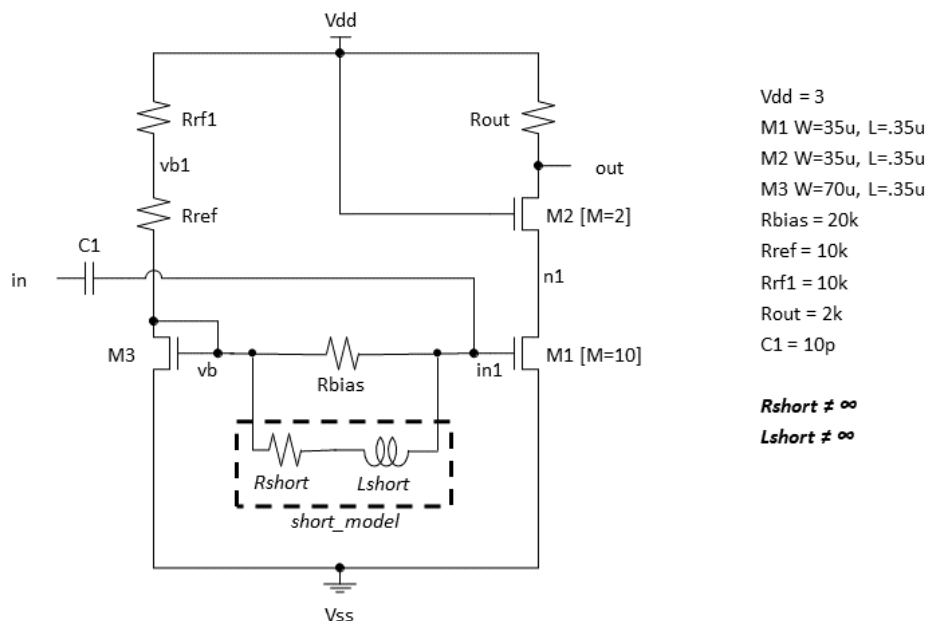


Figure 5—Example of an LNA with a short defect across Rbias

- c) If the defect universe is derived from a circuit's layout, each inter-layer via and sufficiently long and narrow connection in the circuit layout shall be the site of an open defect, and the Defect Coverage Summary shall indicate how potential opens were selected.

NOTE 1—The location of potential open defects can be deduced from graphical analysis of the layout and/or existence of a parasitic resistance in the layout-extracted netlist.

NOTE 2—The term “sufficiently” means greater than a threshold value for the process technology.

- d) If the defect universe is derived from a circuit's layout, each adjacent pair of nodes in a layout shall be the site of a short defect, and the Defect Coverage Summary shall indicate how adjacent nodes were selected.

NOTE 1—Adjacency can be deduced from graphical analysis of the layout and/or existence of a parasitic capacitance in the layout-extracted netlist.

NOTE 2—Adjacency can be on the same layer (see Figure 6) and can be inter-layer.

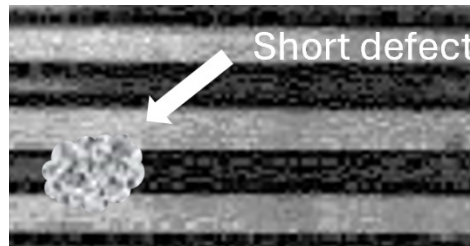


Figure 6—Example of short between two metal interconnects

- e) The Defect Universe Details shall be listed in a syntax that is machine parseable and human readable.
- f) For each circuit element in the simulation netlist of the application circuitry, the Defect Universe Details shall indicate the following:
- 1) Hierarchical instance name of the circuit element
 - 2) Node connected to each terminal of the circuit element, and the terminal type if distinguishable, such as drain, gate, cathode, top, etc.
 - 3) Circuit element type, such as resistor, capacitor, inductor, diode, MOS transistor, bipolar transistor, etc.
 - 4) “No_2427_defect” if the circuit element is to be excluded from consideration for defect injection

NOTE 1—For a two-terminal resistor, capacitor, or inductor, it is not necessary to indicate the terminal type, though the top and bottom plate of a capacitor could be indicated.

NOTE 2—Example reasons for exclusion: modeling artifact, duplicate parasitic, and non-real element, such as a negative resistance.

- g) For each circuit element in the application circuitry, the Defect Universe Details shall indicate the following for each defect associated with the circuit element:
- 1) Serialized identifier
 - 2) Defect type: short, open, or parametric
 - 3) Hierarchical instance name of the circuit element
 - 4) What the defect is applied to:
 - the circuit element terminal name, if the defect is an open
 - the circuit element terminal names, if the defect is a short
 - 5) Defect model name
 - 6) Multiplier, if multiple identical circuit elements are connected in parallel
 - 7) Defect weight

NOTE 1—Circuit element is either a design-intent circuit element or a layout-extracted circuit element.

NOTE 2—A design-intent circuit element could be modeled by a subcircuit netlist that includes multiple elements. Some elements are tagged as `No_2427_defect` and others are design-intent but not subjected to defect injection individually. For example, a design-intent circuit element could be modeled as a distributed series RC network; but only a single parasitic capacitor and single equivalent design-intent resistor are subjected to defect injection (see Figure 7). An open in any of the series resistors in the RC network model is sufficient to represent an open in the original design-intent circuit element.

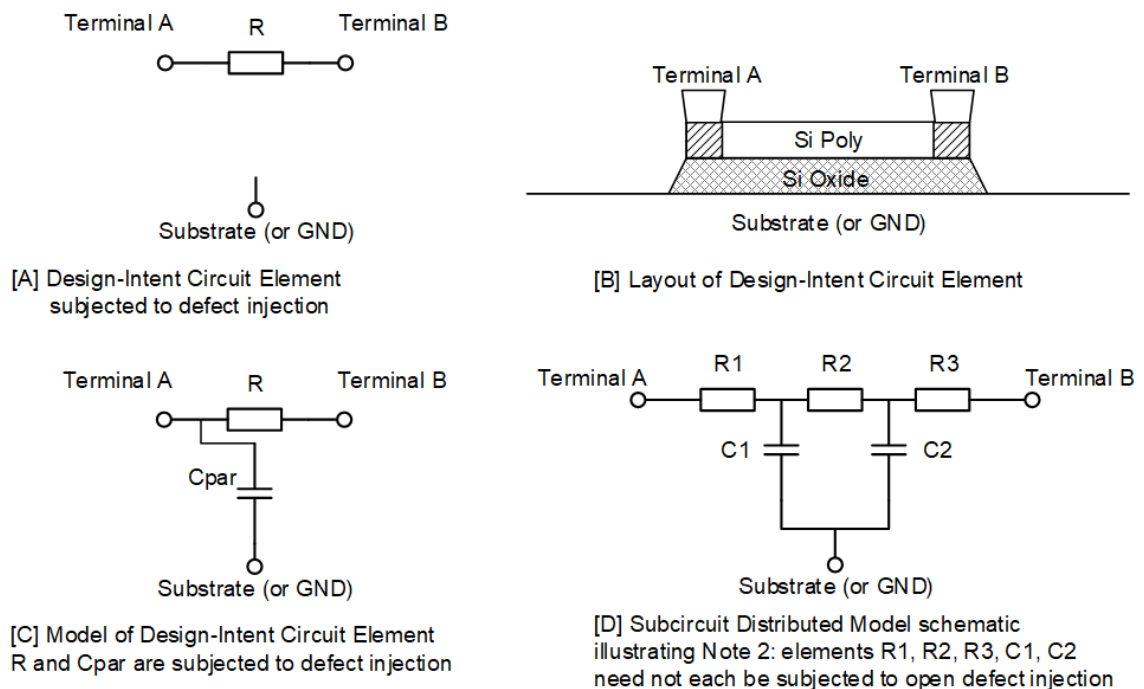


Figure 7—Design-intent element modeled as a 3R-2C network for a poly resistor

- h) For each short defect listed in the Defect Universe Details that is not between two terminals of a single circuit element, the following shall be indicated:
- 1) Serialized identifier

- 2) Defect type: short
 - 3) Nodes connected to the defect model
 - 4) Defect model name
 - 5) Defect weight
- i) For each open defect listed in the Defect Universe Details that is not in a terminal of a circuit element, the following shall be indicated:
- 1) Serialized identifier
 - 2) Defect type: open
 - 3) All terminals of circuit element instances connected to the first terminal of the defect model
 - 4) Defect model name
 - 5) Defect weight

NOTE—The list of terminals connected to the first terminal of the open defect model are provided in the Defect Universe Details for this open. Whereas terminals of circuit elements connected to the second terminal of this open defect model can be derived from the original netlist. Figure 8 shows an example of a node (net1) connected to four terminals (T2, T3, T4, T5), and an open defect injected in the middle such that the first terminal of the open defect connects to T2 and T3, which would be indicated for that defect. Equivalently, T4 and T5 could be indicated, if the defect were injected such that its first terminal connected to T4 and T5 instead.

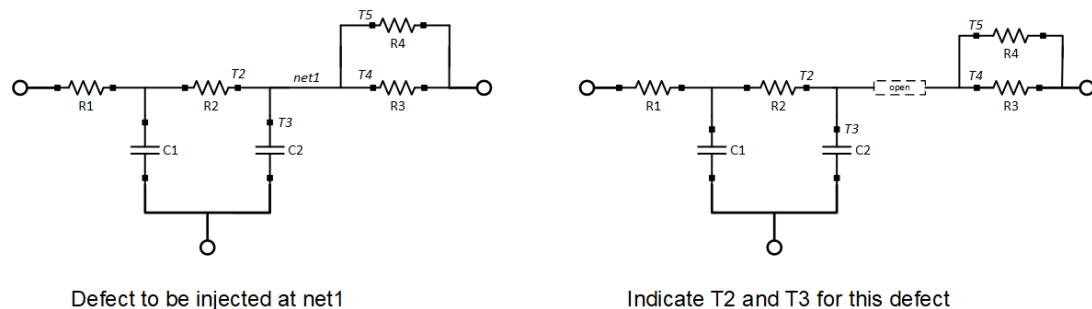


Figure 8—Illustration of Rule 3.1i) defect free circuit and example of the injection of an open defect in the ‘net1’ interconnect, with terminals T2 and T3 to be reported in the Defect Universe Details

- j) For each multi-element parametric defect listed in the Defect Universe Details, the following shall be indicated:
- 1) Serialized identifier
 - 2) Defect type: parametric
 - 3) Hierarchical instance names of the circuit elements
 - 4) Defect model name
 - 5) Defect weight
- k) For each multiple simultaneous independent defect listed in the Defect Universe Details, the following shall be indicated:
- 1) Serialized identifier

- 2) Defect type #1: short, open, or parametric
- 3) Hierarchical instance name of the circuit element
- 4) Defect model name
- 5) Defect weight
- 6) Defect type #n: short, open, or parametric
- 7) Hierarchical instance name of the circuit element
- 8) Defect model name
- 9) Defect weight

NOTE—Multiple defects that are reasonably likely to occur simultaneously are systematic defects. See the notes for 3.5d) and 3.5e) regarding defect weighting when defects are systematic.

- l) If design-intent circuit elements are merged, the Defect Universe Details shall report the resulting merged circuit element with an explicit label “merged,” as well as the number of the parallel design-intent circuit elements that were merged into one circuit element.

NOTE—This rule targets the merging applied as a pre-processing of the application circuit's netlist for the sole purpose of defect universe generation. The commonly used simulation technique of merging of elements during circuit simulation need not be reported.

- m) To indicate that a defect has been intentionally excluded, a weight of “0” shall be used in the Defect Universe Details and an adequate justification shall be provided in the Defect Coverage Summary.

Example 1: In the Defect Coverage Summary, a note like this could be provided to explain why all shorts to the substrate have weight 0: “Since standard cells in this design share a common bulk, opens in the bulk connection are deemed reasonably unlikely and assigned a weight of 0.”

Example 2: In the Defect Coverage Summary, a note like this could be provided to explain why certain shorts have weight 0: “Source-drain shorts are assigned a weight of 0 in this design for transistors with gates longer than 2 μm , since defects larger than 1.5 μm are not considered reasonably likely.”

Recommendations

- n) In the application circuitry netlist, each circuit element should be the site of a potential parametric defect model having a combination of its principal parameters providing excessively high circuit element characteristic performance, and a combination of its principal parameters providing excessively low circuit element characteristic performance.

NOTE 1— An increase in the length, or a decrease in the width, of a resistor would cause an increase in its characteristic performance, resistance (see example in Figure 9). For a diode, a more negative value for its breakdown voltage could be an increase in that characteristic performance.

NOTE 2— A decrease in the area of a capacitor would cause a decrease in its characteristic performance, capacitance. For an NMOS (or PMOS) transistor, a more positive (or negative) threshold voltage could be a decrease in that characteristic performance.

Examples of principal parameters: the resistance, or the length, width, and sheet resistance of a resistor; the capacitance, or the area and oxide thickness of a capacitor; the inductance, or the length and layout radius of an inductor; the saturation current and (negative) reverse breakdown

voltage of a diode; the (positive or negative) threshold voltage, length, width, and oxide thickness of an MOS transistor; the saturation current and beta of a bipolar transistor.

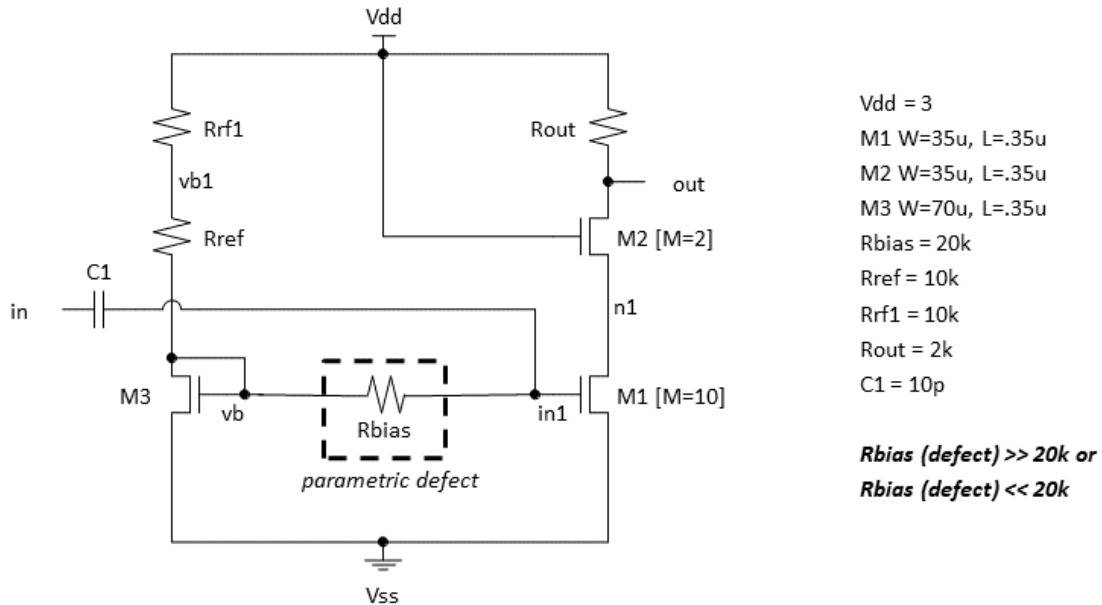


Figure 9—Example of an LNA with a parametric defect in Rbias

- o) Passive elements having resistance, capacitance, or inductance below threshold values specified in the Defect Coverage Summary should be assumed to be parasitic, unless specific tags applied to the given elements indicate otherwise.

NOTE 1—The tagging could be, for example, a text suffix, naming convention, or added parameter setting.

NOTE 2—The specified threshold values could be stated in the Defect Coverage Summary. Parasitics are not design-intent circuit elements. However, they indicate where interconnect opens and shorts could be injected according to rules 3.1h) and 3.1i) and recommendations 3.1p) and 3.1q).

- p) If the space between a node in application circuitry and a node in unused circuitry is small enough that there is a parasitic capacitance between them in a layout-extracted netlist, and there is an intervening power or ground ring, then only a short to that ring should be in the Defect Universe Details.

NOTE—Figure 10 illustrates this recommendation.

- q) If the space between a node in application circuitry and a node in unused circuitry is small enough that there is a parasitic capacitance between them in a layout-extracted netlist, and there is no intervening power or ground ring, then a short between the two nodes should be in the Defect Universe Details.

NOTE—Figure 10 illustrates this recommendation.