# IEEE Standard for the Functional Verification Language *e*

IEEE Computer Society

Developed by the
Design Automation Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

**IEEE Std 1647™-2019**
(Revision of
IEEE Std 1647-2016)

**IEEE Std 1647™-2019**
(Revision of
IEEE Std 1647-2016)

# IEEE Standard for the Functional Verification Language *e*

Developed by the

**Design Automation Standards Committee**
of the
**IEEE Computer Society**

Approved 13 June 2019

**IEEE SA Standards Board**

Grateful acknowledgment is made to Cadence, Inc., for permission to use the following source material:

Namespaces for Types in ***e***, Version 1.1

***e*** Language Reference, Version 18.03, Chapter 3.11 (Defining Interface Types)

***e*** Language Reference, Version 18.03, Chapter 2.13 (The Set Type)

***e*** Language Reference, Version 18.03, Chapter 26 (Set Pseudo Methods)

***e*** Language Reference, Version 18.03, Chapter 5 (Template Type)

***e*** Language Reference, Version 18.03, Chapter 2.1.11 (Defining Custom Numeric Types)

***e*** Language Reference, Version 18.03, Chapter 32 (Tables)

***e*** Language Reference, Version 18.03, Chapter 34 (Annotations)

***e*** Language Reference, Version 18.03, Chapter 28.1 (Interface For Custom Numeric Types)

**Abstract:** The ***e*** *functional verification language* is an application-specific programming language, aimed at automating the task of verifying a hardware or software design with respect to its specification. Verification environments written in ***e*** provide a model of the environment in which the design is expected to function, including the kinds of erroneous conditions the design needs to withstand. A typical verification environment is capable of generating user-controlled test inputs with statistically interesting characteristics. Such an environment can check the validity of the design responses. Functional coverage metrics are used to control the verification effort and gauge the quality of the design. ***e*** verification environments can be used throughout the design cycle, from a high-level architectural model to a fully realized system. A definition of the ***e*** language syntax and semantics and how tool developers and verification engineers should use them are contained in this standard.

**Keywords:** assertion, concurrent programming, constraint, dynamic verification, functional coverage, functional verification, IEEE 1647™, simulation, temporal logic, test generation

---

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

Copyright © 2019 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 9 August 2019. Printed in the United States of America.

Java is a trademark of Sun Microsystems, Inc.

Microsoft and Excel are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Perl is a registered trademark of Perl, Inc.

POSIX is a trademark of The Institute of Electrical and Electronics Engineers, Incorporated.

SystemC is a registered trademark of Open SystemC Initiative.

Verilog is a registered trademark of Cadence Design Systems, Inc.

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

PDF: ISBN 978-1-5044-5977-8 STD23758
Print: ISBN 978-1-5044-5978-5 STDPD23758

*IEEE prohibits discrimination, harassment, and bullying.*
*For more information, see http://www.ieee.org/web/aboutus/whatis/policies/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 documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/ipr/disclaimers.html.

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute 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 Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. 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 his or her 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: PROCUREMENT OF 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.


Copyright © 2019 IEEE. All rights reserved.

## Translations

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

## Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual 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. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

## Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 Standards Coordinating Committees 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 revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

## 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 imply 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.

## 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, IEEE does not waive any rights in copyright to the documents.


Copyright © 2019 IEEE. All rights reserved.

## Photocopies

Subject to payment of the appropriate fee, 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. 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 ten years. When a document is more than ten 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 at http://ieeexplore.ieee.org or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website at http://standards.ieee.org.

## Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

## Patents

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 http://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.


Copyright © 2019 IEEE. All rights reserved.

## Participants

At the time this standard was submitted to the IEEE SA Standards Board for approval, the IEEE ***e*** Functional Verification Language Working Group had the following members and observers:

**Darren Galpin,** *Chair and Technical Editor*
**Yuri Tsoglin,** *Vice-Chair*

Cristian Amitroaie
Mike Bartley
Stefan Birman
Silas McDermott
Genadi Osowiecki
Andrew Piziali
Marcus Harnisch
Yaron Kashai
Efrat Shneydor
Uwe Simm
Akash Singh

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

Robert Aiello
Heiko Ehrenberg
Darren Galpin
Randall Groves
Werner Hoelzl
Osamu Karatsu
Piotr Karocki
Yaron Kashai
Rakesh Kumar
Bansi Patel
Andrew Piziali
Stephen Schwarm
Walter Struppler
John Vergis

When the IEEE SA Standards Board approved this amendment on 13 June 2019, it had the following membership:

**Gary Hoffman,** *Chair*
**Ted Burse,** *Vice Chair*
**Jean-Philippe Faure,** *Past Chair*
**Konstantinos Karachalios,** *Secretary*

Masayuki Ariyoshi
Stephen D. Dukes
J. Travis Griffith
Guido Hiertz
Christel Hunter
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
David J. Law
Joseph Levy
Howard Li
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Annette D. Reilly
Dorothy Stanley
Sha Wei
Phil Wennblom
Philip Winston
Howard Wolfman
Feng Wu
Jingyi Zhou

*Member Emeritus


Copyright © 2019 IEEE. All rights reserved.

# Introduction

This introduction is not part of IEEE Std 1647-2019, IEEE Standard for the Functional Verification Language ***e***.

The ***e*** *functional verification language* is an application-specific programming language aimed at the problem of verifying functional correctness of hardware and software designs. Simply stated, functional verification attempts to provide a quantitative answer to the question: How well does the design match the functional specification?

Functional correctness of chip designs grew in criticality from the mid-1980s. As design complexity kept growing, ad hoc testing methods ran out of steam and a more systematic verification approach was necessary. Manually constructed test suites, the early method of choice, became both uneconomical and ineffective when scaled up. As a result, many companies supplemented manual test suites with *pseudo-random generation* of input stimulus. Such test generation programs were typically built for a particular project or a particular architecture. They turned out to be expensive to develop and maintain, but once functional, they would clean up the design in a very thorough way.

A key observation made by Yoav Hollander, the creator of ***e***, was that verification environments of different projects have a lot in common and yet each verification environment is structured to match a particular design specification. Hollander's solution was to create a language that had verification-specific constructs as primitives and the full capabilities of a high-level language for customization. In particular, pseudo-random test generation became a built-in capability of the language. Early prototypes of the language were experimented with as early as 1993, showing significant productivity gains.

The ***e*** language was productized by Verisity, Ltd., in 1996, as part of a functional verification tool suite. The proliferation of the ***e*** language and the growing investment in ***e***-based intellectual property (IP) compelled the creation of the ***e*** *steering committee* in June of 2002, composed of individuals from Texas Instruments, Rambus, ST Microelectronics, Cisco, Intel, Axis System, STARC, and Verisity. The ***e*** *steering committee* recommended the ***e*** language be standardized through The Institute of Electrical and Electronics Engineers (IEEE). Accepting the recommendation, Verisity released the rights to the language to the IEEE in June of 2003.

The ***e*** language, in its current form, brings together concepts from many domains.

- ***e*** has a basic object-oriented (OO) programming model, with implicit memory management and single inheritance. In this, ***e*** is similar to Java™.[a]
- Beyond objects, ***e*** supports *aspects*, which can be viewed as layers cutting across multiple objects. Adding an aspect to an existing program refines the program by introducing a coherent change to a plurality of objects.
- ***e*** supports constraints as object features. Constraints are used to refine object modeling. The execution model of the language involves resolving constraint systems and picking random values that would satisfy constraint systems.
- ***e*** is a strongly typed language, like Pascal and Modula.
- ***e*** has concurrency constructs and modeling blocks for hierarchical composition, similar to hardware description languages like Verilog®[b](see IEEE Std 1364™)[c] and VHDL (see IEC/IEEE 61691-1-1). Concurrency in ***e*** is synchronous, like in Esterel.
- ***e*** contains a temporal language that borrows from Linear Temporal Logic and Interval Temporal Logic.

[a] Java is a trademark of Sun Microsystems, Inc. in the United States and other countries.

[b] Verilog is a registered trademark of Cadence Design Systems, Inc.

[c] Information on references can be found in Clause 2.



Copyright © 2019 IEEE. All rights reserved.

— ***e*** has many built-in constructs aimed at simplifying common programming tasks; built-in support for lists and hashes; and pattern matching and string- and file-manipulation features, which are borrowed from Perl®.[d]

— The ***e*** syntax is extendable with a powerful macro capability.

This combination of concepts caters directly to the needs of verification engineers, removing the need to cobble together multiple components in different languages.

As with any programming language, the source of ingenuity is with the programmer. Verification engineers need sound methodologies, creativity, an inquisitive mind, and a keen eye for poorly specified aspects. Yet experience with ***e*** shows that when put to good use, the ***e*** language fosters productivity and quality results.

[d] Perl is a registered trademark of Perl, Inc.


Copyright © 2019 IEEE. All rights reserved.

# Contents




Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

Copyright © 2019 IEEE. All rights reserved.

# IEEE Standard for the Functional Verification Language *e*

## 1. Overview

### 1.1 Introduction

This clause explains the scope and purpose of this standard; gives an overview of the basic concepts, major semantic components, and conventions used in this standard; and summarizes its contents.

### 1.2 Scope

This standard defines the ***e*** functional verification language. This standard aims to serve as an authoritative source for the definition of (a) syntax and semantics of ***e*** language constructs (b) the ***e*** language interaction with standard simulation languages (c) ***e*** language libraries.

### 1.3 Purpose

This standard serves the community involved with functional verification of electronic designs using the ***e*** language. It provides an implementation independent definition of the ***e*** language and facilitates the development of ***e*** language based design automation tools.

### 1.4 Verification environments

*Electronic systems* are integrated circuits (ICs), boards, or modules combining multiple ICs together, along with optional embedded processors and software components. Electronic systems are built to *specifications* that anticipate the environment in which such systems are expected to function and define the expected system functionality. *Functional verification* measures how well a system meets its specification. Even with moderately complex systems this question cannot be answered by inspection. For all modern electronic systems, a sophisticated *verification process* needs to accompany the design process to ensure compliance with the specification.

Many electronic design automation (EDA) tools are used to carry out the functional verification process. The most prominent functional verification method, used to verify virtually all system designs, is called *dynamic verification* or *simulation-based verification*. Simulation-based verification makes use of a functional model of the system being designed. The functional model is *simulated* in the context of a mock-up of the anticipated system environment. This mock-up is called the *verification environment*.


Copyright © 2019 IEEE. All rights reserved.