IEC 62530

Edition 3.0 2021-07

INTERNATIONAL STANDARD

IEEE Std 1800™

SystemVerilog – Unified Hardware Design, Specification, and Verification Language

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.01; 35.060

ISBN 978-2-8322-9977-7

Contents

Part One: Design and Verification Constructs

Part Two: Hierarchy Constructs

Part Three: Application Programming Interfaces

List of figures

List of tables

List of syntax excerpts

SystemVerilog – Unified Hardware Design, Specification, and Verification Language

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC document(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

   IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. 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 contained in its standards. Use of IEEE Standards documents is wholly voluntary. *IEEE documents are made available for use subject to important notices and legal disclaimers (see* https://standards.ieee.org/ipr/disclaimers.html *for more information).*

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations. This Dual Logo International Standard was jointly developed by the IEC and IEEE under the terms of that agreement.

2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent 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.

IEC 62530/IEEE Std 1800 was processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement. It is an International Standard.

The text of this International Standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1800 (2017)	91/1714/FDIS	91/1726/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IEEE Std 1800™-2017
(Revision of
IEEE Std 1800-2012)

IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language

Sponsor

Design Automation Standards Committee
of the
IEEE Computer Society
and the
IEEE Standards Association Corporate Advisory Group

Approved 6 December 2017

IEEE-SA Standards Board

Abstract: The definition of the language syntax and semantics for SystemVerilog, which is a unified hardware design, specification, and verification language, is provided. This standard includes support for modeling hardware at the behavioral, register transfer level (RTL), and gate-level abstraction levels, and for writing testbenches using coverage, assertions, object-oriented programming, and constrained random verification. The standard also provides application programming interfaces (APIs) to foreign programming languages.

Keywords: assertions, design automation, design verification, hardware description language, HDL, HDVL, IEEE 1800™, PLI, programming language interface, SystemVerilog, Verilog®, VPI

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.

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.

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 the IEEE-SA Website 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.

Introduction

This introduction is not part of IEEE Std 1800-2017, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

The purpose of this standard is to provide the electronic design automation (EDA), semiconductor, and system design communities with a well-defined and official IEEE unified hardware design, specification, and verification standard language. The language is designed to coexist and enhance the hardware description and verification languages (HDVLs) presently used by designers while providing the capabilities lacking in those languages.

SystemVerilog is a unified hardware design, specification, and verification language based on the Accellera SystemVerilog 3.1a extensions to the Verilog hardware description language (HDL) [B4], published in 2004. Accellera is a consortium of EDA, semiconductor, and system companies. IEEE Std 1800 enables a productivity boost in design and validation and covers design, simulation, validation, and formal assertion-based verification flows.

SystemVerilog enables the use of a unified language for abstract and detailed specification of the design, specification of assertions, coverage, and testbench verification based on manual or automatic methodologies. SystemVerilog offers application programming interfaces (APIs) for coverage and assertions, and a direct programming interface (DPI) to access proprietary functionality. SystemVerilog offers methods that allow designers to continue to use present design languages when necessary to leverage existing designs and intellectual property (IP). This standardization project will provide the VLSI design engineers with a well-defined IEEE standard, which meets their requirements in design and validation, and which enables a step function increase in their productivity. This standardization project will also provide the EDA industry with a standard to which they can adhere and that they can support in order to deliver their solutions in this area.

Part One:
Design and Verification Constructs

IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language

1. Overview

1.1 Scope

This standard provides the definition of the language syntax and semantics for the IEEE 1800™ SystemVerilog language, which is a unified hardware design, specification, and verification language. The standard includes support for behavioral, register transfer level (RTL), and gate-level hardware descriptions; testbench, coverage, assertion, object-oriented, and constrained random constructs; and also provides application programming interfaces (APIs) to foreign programming languages.

1.2 Purpose

This standard develops the IEEE 1800 SystemVerilog language in order to meet the increasing usage of the language in specification, design, and verification of hardware. This revision corrects errors and clarifies aspects of the language definition in IEEE Std 1800-2012.[1] This revision also provides enhanced features that ease design, improve verification, and enhance cross-language interactions.

1.3 Content summary

This standard serves as a complete specification of the SystemVerilog language. This standard contains the following:

- The formal syntax and semantics of all SystemVerilog constructs
- Simulation system tasks and system functions, such as text output display commands
- Compiler directives, such as text substitution macros and simulation time scaling
- The Programming Language Interface (PLI) mechanism
- The formal syntax and semantics of the SystemVerilog Verification Procedural Interface (VPI)
- An Application Programming Interface (API) for coverage access not included in VPI

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

— Direct programming interface (DPI) for interoperation with the C programming language
— VPI, API, and DPI header files
— Concurrent assertion formal semantics
— The formal syntax and semantics of standard delay format (SDF) constructs
— Informative usage examples

NOTE—An earlier standard, IEEE Std 1800-2009, represented a merger of two previous standards: IEEE Std 1364™-2005 and IEEE Std 1800-2005. In these previous standards, Verilog® was the base language and defined a completely self-contained standard. SystemVerilog defined a number of significant extensions to Verilog, but IEEE Std 1800-2005 was not a self-contained standard; IEEE Std 1800-2005 referred to, and relied on, IEEE Std 1364-2005. These two standards were designed to be used as one language. Merging the base Verilog language into the SystemVerilog standard enabled users to have all information regarding syntax and semantics in a single document.[2, 3]

1.4 Special terms

Throughout this standard, the following terms apply:

— *SystemVerilog 3.1a* refers to the Accellera *SystemVerilog 3.1a Language Reference Manual* [B4], a precursor to IEEE Std 1800-2005.[4]
— *Verilog* refers to IEEE Std 1364-2005 for the Verilog hardware description language (HDL).
— *Language Reference Manual* (*LRM*) refers to the document describing a Verilog or SystemVerilog standard.
— *Tool* refers to a software implementation that reads SystemVerilog source code, such as a logic simulator.

NOTE—In IEEE Std 1800-2005, *SystemVerilog* referred to just the extensions to the IEEE 1364-2005 Verilog language and did not include the Verilog base language.

1.5 Conventions used in this standard

This standard is organized into clauses, each of which focuses on a specific area of the language. There are subclauses within each clause to discuss individual constructs and concepts. The discussion begins with an introduction and an optional rationale for the construct or the concept, followed by syntax and semantic descriptions, followed by examples and notes.

The terminology conventions used throughout this standard are as follows:

— The word *shall* is used to indicate 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*).
— The word *should* is used to indicate 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; or that (in the negative form) a certain course of action is deprecated but not prohibited (*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*).

[2]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.
[3]Verilog is a registered trademark of Cadence Design Systems, Inc.
[4]The numbers in brackets correspond to those of the bibliography in Annex Q.

1.6 Syntactic description

The main text uses the following conventions:

— *Italicized* font when a term is being defined

— `Constant-width` font for examples, file names, and references to constants, especially `0`, `1`, `x`, and `z` values

— **`Boldface constant-width`** font for SystemVerilog keywords, when referring to the actual keyword

The formal syntax of SystemVerilog is described using Backus-Naur Form (BNF). The following conventions are used:

— Lowercase words, some containing embedded underscores, denote syntactic categories. For example:

```
module_declaration
```

— **Boldface-red** characters denote reserved keywords, operators, and punctuation marks as a required part of the syntax. For example:

```
module   =>   ;
```

— A vertical bar (|) that is not in boldface-red separates alternative items. For example:

```
unary_operator ::=
      + | - | ! | ~ | & | ~& | | | ~| | ^ | ~^ | ^~
```

— Square brackets ([ ]) that are not in boldface-red enclose optional items. For example:

```
function_declaration ::= function [ lifetime ] function_body_declaration
```

— Braces ({ }) that are not in boldface-red enclose a repeated item. The item may appear zero or more times; the repetitions occur from left to right as with an equivalent left-recursive rule. Thus, the following two rules are equivalent:

```
list_of_param_assignments ::= param_assignment { , param_assignment }
list_of_param_assignments ::=
      param_assignment
    | list_of_param_assignments , param_assignment
```

A *qualified term* in the syntax is a term such as *array_identifier* for which the “array” portion represents some semantic intent and the “identifier” term indicates that the qualified term reduces to the “identifier” term in the syntax. The syntax does not completely define the semantics of such qualified terms; for example while an identifier that would qualify semantically as an array_identifier is created by a declaration, such declaration forms are not explicitly described using *array_identifier* in the syntax.

1.7 Use of color in this standard

This standard uses a minimal amount of color to enhance readability. The coloring is not essential and does not affect the accuracy of this standard when viewed in pure black and white. The places where color is used are the following:

— Cross references that are hyperlinked to other portions of this standard are shown in underlined-blue text (hyperlinking works when this standard is viewed interactively as a PDF file).

— Syntactic keywords and tokens in the formal language definitions are shown in **`boldface-red text`**.

— Some figures use a minimal amount of color to enhance readability.

1.8 Contents of this standard

A synopsis of the clauses and annexes is presented as a quick reference. All clauses and several of the annexes are normative parts of this standard. Some annexes are included for informative purposes only.

Part One: Design and Verification Constructs

Clause 1 describes the contents of this standard and the conventions used in this standard.

Clause 2 lists references to other standards that are required in order to implement this standard.

Clause 3 introduces the major building blocks that make up a SystemVerilog design and verification environment: modules, programs, interfaces, checkers, packages, and configurations. This clause also discusses primitives, name spaces, the `$unit` compilation space, and the concept of simulation time.

Clause 4 describes the SystemVerilog simulation scheduling semantics.

Clause 5 describes the lexical tokens used in SystemVerilog source text and their conventions.

Clause 6 describes SystemVerilog data objects and types, including nets and variables, their declaration syntax and usage rules, and charge strength of the values on nets. This clause also discusses strings and string methods, enumerated types, user-defined types, constants, data scope and lifetime, and type compatibility.

Clause 7 describes SystemVerilog compound data types: structures, unions, arrays, including packed and unpacked arrays, dynamic arrays, associative arrays, and queues. This clause also describes various array methods.

Clause 8 describes the object-oriented programming capabilities in SystemVerilog. Topics include defining classes, interface classes, dynamically constructing objects, inheritance and subclasses, data hiding and encapsulation, polymorphism, and parameterized classes.

Clause 9 describes the SystemVerilog procedural blocks: **`initial`**, **`always`**, **`always_comb`**, **`always_ff`**, **`always_latch`**, and **`final`**. Sequential and parallel statement grouping, block names, statement labels, and process control are also described.

Clause 10 describes continuous assignments, blocking and nonblocking procedural assignments, and procedural continuous assignments.

Clause 11 describes the operators and operands that can be used in expressions.

Clause 12 describes SystemVerilog procedural programming statements, such as decision statements and looping constructs.

Clause 13 describes tasks and functions, which are subroutines that can be called from more than one place in a behavioral model.

Clause 14 defines clocking blocks, input and output skews, cycle delays, and default clocking.

Clause 15 describes interprocess communications using event types and event controls, and built-in semaphore and mailbox classes.

Clause 16 describes immediate and concurrent assertions, properties, sequences, sequence operations, multiclock sequences, and clock resolution.

Clause 17 describes checkers. Checkers allow the encapsulation of assertions and modeling code to create a single verification entity.

Clause 18 describes generating random numbers, constraining random number generation, dynamically changing constraints, seeding random number generators (RNGs), and randomized **`case`** statement execution.

Clause 19 describes coverage groups, coverage points, cross coverage, coverage options, and coverage methods.

Clause 20 describes most of the built-in system tasks and system functions.

Clause 21 describes additional system tasks and system functions that are specific to input/output (I/O) operations.

Clause 22 describes various compiler directives, including a directive for controlling reserved keyword compatibility between versions of previous Verilog and SystemVerilog standards.

Part Two: Hierarchy Constructs

Clause 23 describes how hierarchies are created in SystemVerilog using module instances and interface instances, and port connection rules. This clause also discusses the `$root` top-level instances, nested modules, extern modules, identifier search rules, how parameter values can be overridden, and binding auxiliary code to scopes or instances.

Clause 24 describes the testbench program construct, the elimination of testbench race conditions, and program control tasks.

Clause 25 describes interface syntax, interface ports, modports, interface subroutines, parameterized interfaces, virtual interfaces, and accessing objects within interfaces.

Clause 26 describes user-defined packages and the std built-in package.

Clause 27 describes the generate construct and how generated constructs can be used to do conditional or multiple instantiations of procedural code or hierarchy.

Clause 28 describes the gate- and switch-level primitives and logic strength modeling.

Clause 29 describes how a user-defined primitive (UDP) can be defined and how these primitives are included in SystemVerilog models.

Clause 30 describes how to specify timing relationships between input and output ports of a module.

Clause 31 describes how timing checks are used in specify blocks to determine whether signals obey the timing constraints.

Clause 32 describes the syntax and semantics of SDF constructs.

Clause 33 describes how to configure the contents of a design.

Clause 34 describes encryption and decryption of source text regions.

Part Three: Application Programming Interfaces

Clause 35 describes SystemVerilog's direct programming interface (DPI), a direct interface to foreign languages and the syntax for importing functions from a foreign language and exporting subroutines to a foreign language.

Clause 36 provides an overview of the programming language interface (PLI and VPI).

Clause 37 presents the VPI data model diagrams, which document the VPI object relationships and access methods.

Clause 38 describes the VPI routines.

Clause 39 describes the assertion API in SystemVerilog.

Clause 40 describes the coverage API in SystemVerilog.

Part Four: Annexes

Annex A (normative) defines the formal syntax of SystemVerilog, using BNF.

Annex B (normative) lists the SystemVerilog keywords.

Annex C (informative) lists constructs that have been deprecated from SystemVerilog. The annex also discusses the possible deprecation of the **`defparam`** statement and the procedural **`assign`**/**`deassign`** statements.

Annex D (informative) describes system tasks and system functions that are frequently used, but that are not required in this standard.

Annex E (informative) describes compiler directives that are frequently used, but that are not required in this standard.

Annex F (normative) describes a formal semantics for SystemVerilog concurrent assertions.

Annex G (normative) describes the SystemVerilog standard package, containing type definitions for mailbox, semaphore, randomize, and process.

Annex H (normative) defines the C language layer for the SystemVerilog DPI.

Annex I (normative) defines the standard `svdpi.h` include file for use with SystemVerilog DPI applications.

Annex J (normative) describes common guidelines for the inclusion of foreign language code into a SystemVerilog application.

Annex K (normative) provides a listing of the contents of the `vpi_user.h` file.

Annex L (normative) provides a listing of the contents of the `vpi_compatibility.h` file, which extends the `vpi_user.h` include file.

Annex M (normative) provides a listing of the contents of the `sv_vpi_user.h` file, which extends the `vpi_user.h` include file.

Annex N (normative) provides the C source code for the SystemVerilog random distribution system functions.

Annex O (informative) describes various scenarios that can be used for intellectual property (IP) protection, and it also shows how the relevant pragmas can be used to achieve the desired effect of securely protecting, distributing, and decrypting the model.

Annex P (informative) defines terms that are used in this standard.

Annex Q (informative) lists reference documents that are related to this standard.

1.9 Deprecated clauses

Annex C lists constructs that appeared in previous versions of either IEEE Std 1364 or IEEE Std 1800, but that have been deprecated and do not appear in this standard. This annex also lists constructs that appear in this standard, but that are under consideration for deprecation in a future version of this standard.

1.10 Examples

Small SystemVerilog code examples are shown throughout this standard. These examples are informative. They are intended to illustrate the usage of SystemVerilog constructs in a simple context and do not define the full syntax.

1.11 Prerequisites

Some clauses of this standard presuppose a working knowledge of the C programming language.

2. Normative references

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used, so each referenced document is cited in the text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

Anderson, R., Biham, E., and Knudsen, L. “Serpent: A Proposal for the Advanced Encryption Standard,” NIST AES Proposal, 1998.[5]

ANSI X9.52-1998, American National Standard for Financial Services—Triple Data Encryption Algorithm Modes of Operation.[6]

ElGamal, T., “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469–472, July 1985.

FIPS 46-3 (October 1999), Data Encryption Standard (DES).[7]

FIPS 180-2 (August 2002), Secure Hash Standard (SHS).

FIPS 197 (November 2001), Advanced Encryption Standard (AES).

IEEE Std 754™, IEEE Standard for Floating-Point Arithmetic.[8, 9]

IEEE Std 1003.1™, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®).

IEEE Std 1364™-1995, IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language.

IEEE Std 1364™-2001, IEEE Standard Verilog Hardware Description Language.

IEEE Std 1364™-2005, IEEE Standard for Verilog Hardware Description Language.

IEEE Std 1800™-2005, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IEEE Std 1800™-2009, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IEEE Std 1800™-2012, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IETF RFC 1319 (April 1992), The MD2 Message-Digest Algorithm.[10]

IETF RFC 1321 (April 1992), The MD5 Message-Digest Algorithm.

IETF RFC 2045 (November 1996), Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies.

---

[5]This document is available at http://www.cl.cam.ac.uk/~rja14/Papers/serpent.tar.gz.

[6]ANSI publications are available from the American National Standards Institute (http://www.ansi.org/).

[7]FIPS publications are available from the National Technical Information Service (http://www.ntis.gov/).

[8]IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[9]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[10]IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

IETF RFC 2144 (May 1997), The CAST-128 Encryption Algorithm.

IETF RFC 2437 (October 1998), PKCS #1: RSA Cryptography Specifications, Version 2.0.

IETF RFC 2440 (November 1998), OpenPGP Message Format.

ISO/IEC 10118-3:2004, Information technology—Security techniques—Hash-functions—Part 3: Dedicated hash-functions.[11]

Schneier, B., "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993)*, Springer-Verlag, 1994, pp. 191–204.

Schneier, B., et al., *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*, 1st ed., Wiley, 1999.

[11]ISO publications are available from the International Organization for Standardization (http://www.iso.org/). IEC publications are available from the International Electrotechnical Commission (http://www.iec.ch). ISO/IEC publications are available from the American National Standards Institute (http://www.ansi.org/)..