



IEC 62530

Edition 1.0 2007-11

INTERNATIONAL STANDARD

IEEE 1800™

**Standard for SystemVerilog – Unified Hardware Design, Specification, and
Verification Language**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE

XT

ICS 25.040

ISBN 2-8318-9349-6

CONTENTS

IEEE introduction.....	10
FOREWORD	13
1. Overview.....	14
1.1 Scope.....	14
1.2 Purpose.....	14
1.3 Conventions used in this standard	16
1.4 Syntactic description.....	16
1.5 Use of color in this standard	17
1.6 Contents of this standard.....	17
1.7 Examples.....	20
1.8 Prerequisites.....	20
2. Normative references.....	22
3. Literal values.....	24
3.1 Introduction.....	24
3.2 Literal value syntax.....	24
3.3 Integer and logic literals	25
3.4 Real literals	25
3.5 Time literals	25
3.6 String literals.....	25
3.7 Array literals	26
3.8 Structure literals.....	26
4. Data types	22
4.1 Introduction.....	28
4.2 Data type syntax.....	29
4.3 Integer data types	30
4.4 Real and shortreal data types	31
4.5 Void data type.....	31
4.6 Chandle data type.....	31
4.7 String data type	32
4.8 Event data type.....	37
4.9 User-defined types	37
4.10 Enumerations	39
4.11 Structures and unions.....	44
4.12 Class.....	50
4.13 Singular and aggregate types	50
4.14 Casting	50
4.15 \$cast dynamic casting	51
4.16 Bit-stream casting	52
4.17 Default attribute type	55
5. Arrays.....	56
5.1 Introduction.....	56
5.2 Packed and unpacked arrays	56
5.3 Multiple dimensions	57
5.4 Indexing and slicing of arrays.....	58
5.5 Array querying functions	59
5.6 Dynamic arrays	59
5.7 Array assignment	61
5.8 Arrays as arguments.....	62
5.9 Associative arrays	63

5.10	Associative array methods	66
5.11	Associative array assignment.....	68
5.12	Associative array arguments	69
5.13	Associative array literals.....	69
5.14	Queues	69
5.15	Array manipulation methods	72
6.	Data declarations.....	78
6.1	Introduction.....	78
6.2	Data declaration syntax.....	78
6.3	Constants.....	79
6.4	Variables	83
6.5	Nets	84
6.6	Scope and lifetime	85
6.7	Nets, regs, and logic.....	86
6.8	Signal aliasing.....	87
6.9	Type compatibility	89
6.10	Type operator	92
7.	Classes	94
7.1	Introduction.....	94
7.2	Syntax	94
7.3	Overview.....	95
7.4	Objects (class instance).....	96
7.5	Object properties.....	96
7.6	Object methods	97
7.7	Constructors	97
7.8	Static class properties.....	98
7.9	Static methods.....	99
7.10	This	99
7.11	Assignment, renaming, and copying.....	100
7.12	Inheritance and subclasses	101
7.13	Overridden members.....	101
7.14	Super	102
7.15	Casting	103
7.16	Chaining constructors	103
7.17	Data hiding and encapsulation	104
7.18	Constant class properties	104
7.19	Abstract classes and virtual methods	105
7.20	Polymorphism: dynamic method lookup.....	106
7.21	Class scope resolution operator ::	106
7.22	Out-of-block declarations	107
7.23	Parameterized classes	108
7.24	Typedef class	109
7.25	Classes and structures	110
7.26	Memory management	110
8.	Operators and expressions	112
8.1	Introduction.....	112
8.2	Operator syntax.....	112
8.3	Assignment operators	114
8.4	Operations on logic and bit types	114
8.5	Wild equality and wild inequality.....	115
8.6	Real operators	115

8.7	Size.....	116
8.8	Sign	116
8.9	Operator precedence and associativity	116
8.10	Built-in methods	116
8.11	Static prefixes	117
8.12	Concatenation	118
8.13	Assignment patterns.....	119
8.14	Tagged union expressions and member access.....	124
8.15	Aggregate expressions	125
8.16	Operator overloading	125
8.17	Streaming operators (pack/unpack)	127
8.18	Conditional operator	131
8.19	Set membership.....	131
9.	Scheduling semantics.....	134
9.1	Execution of a hardware model and its verification environment	134
9.2	Event simulation	134
9.3	The stratified event scheduler	134
9.4	The PLI callback control points.....	138
10.	Procedural statements and control flow	140
10.1	Introduction.....	140
10.2	Statements.....	140
10.3	Blocking and nonblocking assignments	141
10.4	Selection statements.....	142
10.5	Loop statements	149
10.6	Jump statements	151
10.7	Final blocks.....	151
10.8	Named blocks and statement labels	152
10.9	Disable	153
10.10	Event control.....	154
10.11	Level-sensitive sequence controls	156
10.12	Procedural assign and deassign removal	156
11.	Processes.....	158
11.1	Introduction.....	158
11.2	Combinational logic.....	158
11.3	Latched logic.....	159
11.4	Sequential logic.....	159
11.5	Continuous assignments	159
11.6	fork..join.....	160
11.7	Process execution threads	161
11.8	Process control.....	161
11.9	Fine-grain process control	163
12.	Tasks and functions	166
12.1	Introduction.....	166
12.2	Tasks	166
12.3	Functions.....	168
12.4	Task and function argument passing	170
12.5	Import and export functions.....	173
13.	Random constraints.....	176
13.1	Introduction.....	176

13.2	Overview.....	176
13.3	Random variables	179
13.4	Constraint blocks	181
13.5	Randomization methods	194
13.6	In-line constraints—randomize() with.....	197
13.7	Disabling random variables with rand_mode()	197
13.8	Controlling constraints with constraint_mode()	198
13.9	Dynamic constraint modification.....	199
13.10	In-line random variable control	200
13.11	Randomization of scope variables—std::randomize().....	201
13.12	Random number system functions and methods	202
13.13	Random stability	204
13.14	Manually seeding randomize	206
13.15	Random weighted case—randcase	207
13.16	Random sequence generation—randsequence.....	208
14.	Interprocess synchronization and communication.....	216
14.1	Introduction.....	216
14.2	Semaphores.....	216
14.3	Mailboxes.....	217
14.4	Parameterized mailboxes	220
14.5	Event	221
14.6	Event sequencing: wait_order()	223
14.7	Event variables.....	224
15.	Clocking blocks	226
15.1	Introduction.....	226
15.2	Clocking block declaration	226
15.3	Input and output skews	228
15.4	Hierarchical expressions	229
15.5	Signals in multiple clocking blocks.....	229
15.6	Clocking block scope and lifetime.....	229
15.7	Multiple clocking blocks example.....	230
15.8	Interfaces and clocking blocks.....	230
15.9	Clocking block events.....	232
15.10	Cycle delay: ##	232
15.11	Default clocking.....	233
15.12	Input sampling	234
15.13	Synchronous events	234
15.14	Synchronous drives.....	235
16.	Program block.....	238
16.1	Introduction.....	238
16.2	The program construct	238
16.3	Eliminating testbench races	240
16.4	Blocking tasks in cycle/event mode.....	241
16.5	Programwide space and anonymous programs.....	241
16.6	Program control tasks	242
17.	Assertions.....	244
17.1	Introduction.....	244
17.2	Immediate assertions.....	244
17.3	Concurrent assertions overview	246
17.4	Boolean expressions	247

17.5	Sequences.....	249
17.6	Declaring sequences	252
17.7	Sequence operations	255
17.8	Manipulating data in a sequence.....	262
17.9	Calling subroutines on match of a sequence.....	276
17.10	System functions.....	277
17.11	Declaring properties.....	277
17.12	Multiclock support.....	290
17.13	Concurrent assertions.....	298
17.14	Clock resolution.....	304
17.15	Binding properties to scopes or instances.....	310
17.16	Expect statement.....	312
17.17	Clocking blocks and concurrent assertions.....	313
18.	Coverage	316
18.1	Introduction.....	316
18.2	Defining the coverage model: covergroup.....	316
18.3	Using covergroup in classes	319
18.4	Defining coverage points.....	321
18.5	Defining cross coverage.....	327
18.6	Specifying coverage options.....	331
18.7	Predefined coverage methods	336
18.8	Predefined coverage system tasks and functions.....	337
18.9	Organization of option and type_option members	337
18.10	Coverage computation	338
19.	Hierarchy	340
19.1	Introduction.....	340
19.2	Packages.....	340
19.3	Compilation unit support	345
19.4	Top-level instance.....	346
19.5	Module declarations.....	347
19.6	Nested modules.....	347
19.7	Extern modules	349
19.8	Port declarations	350
19.9	List of port expressions.....	351
19.10	Time unit and precision	339
19.11	Module instances	353
19.12	Port connection rules	357
19.13	Name spaces	359
19.14	Hierarchical names	360
20.	Interfaces.....	362
20.1	Introduction.....	362
20.2	Interface syntax.....	363
20.3	Ports in interfaces.....	367
20.4	Modports.....	368
20.5	Interfaces and specify blocks.....	374
20.6	Tasks and functions in interfaces.....	375
20.7	Parameterized interfaces.....	381
20.8	Virtual interfaces.....	383
20.9	Access to interface objects.....	387
21.	Configuration libraries.....	390

21.1	Introduction.....	390
21.2	Libraries	390
22.	System tasks and system functions	392
22.1	Introduction.....	392
22.2	Type name function	392
22.3	Expression size system function	393
22.4	Range system function.....	393
22.5	Shortreal conversions.....	394
22.6	Array querying system functions	394
22.7	Assertion severity system tasks	396
22.8	Assertion control system tasks.....	397
22.9	Assertion system functions	397
22.10	Random number system functions.....	398
22.11	Program control	398
22.12	Coverage system functions	398
22.13	Enhancements to Verilog system tasks.....	398
22.14	\$readmemb and \$readmemh.....	400
22.15	\$writememb and \$writememh	400
22.16	File format considerations for multidimensional unpacked arrays.....	401
22.17	System task arguments for multidimensional unpacked arrays.....	402
23.	Compiler directives.....	404
23.1	Introduction.....	404
23.2	`define macros.....	404
23.3	`include	405
23.4	`begin_keywords and `end_keywords	405
24.	Value change dump (VCD) data.....	408
24.1	Introduction.....	408
24.2	VCD extensions	408
25.	Deprecated constructs	410
25.1	Introduction.....	410
25.2	Defparam statements.....	410
25.3	Procedural assign and deassign statements.....	410
26.	Direct programming interface (DPI).....	412
26.1	Overview.....	412
26.2	Two layers of the DPI.....	413
26.3	Global name space of imported and exported functions.....	414
26.4	Imported tasks and functions	415
26.5	Calling imported functions	421
26.6	Exported functions.....	423
26.7	Exported tasks.....	423
26.8	Disabling DPI tasks and functions.....	424
27.	SystemVerilog VPI object model	426
27.1	Introduction.....	426
27.2	Module (supersedes 26.6.1 of IEEE Std 1364).....	428
27.3	Interface	429
27.4	Modport	429
27.5	Interface task and function declaration.....	429
27.6	Program.....	430

27.7	Instance	431
27.8	Instance arrays (supersedes 26.6.2 of IEEE Std 1364)	432
27.9	Scope (supersedes 26.6.3 of IEEE Std 1364)	433
27.10	IO declaration (supersedes 26.6.4 of IEEE Std 1364)	434
27.11	Ports (supersedes 26.6.5 of IEEE Std 1364)	435
27.12	Reference objects	436
27.13	Nets (supersedes 26.6.6 of IEEE Std 1364)	440
27.14	Variables (supersedes 26.6.7 and 26.6.8 of IEEE Std 1364)	443
27.15	Variable select (supersedes 26.6.8 of IEEE Std 1364)	446
27.16	Variable drivers and loads (supersedes 26.6.23 of IEEE Std 1364)	446
27.17	Typespec	447
27.18	Structures and unions	448
27.19	Named events (supersedes 26.6.11 of IEEE Std 1364)	449
27.20	Parameter (supersedes 26.6.12 of IEEE Std 1364)	450
27.21	Class definition	451
27.22	Class variables and class objects	452
27.23	Constraint, constraint ordering, distribution	454
27.24	Constraint expression	455
27.25	Module path, path term (supersedes 26.6.15 of IEEE Std 1364)	455
27.26	Task and function declaration (supersedes 26.6.18 of IEEE Std 1364)	456
27.27	Task and function call (supersedes 26.6.19 of IEEE Std 1364)	457
27.28	Frames (supersedes 26.6.20 of IEEE Std 1364)	458
27.29	Threads	459
27.30	Clocking block	460
27.31	Assertion	461
27.32	Concurrent assertions	462
27.33	Property declaration	462
27.34	Property specification	463
27.35	Sequence declaration	464
27.36	Sequence expression	465
27.37	Multiclock sequence expression	466
27.38	Simple expressions (supersedes 26.6.25 of IEEE Std 1364)	467
27.39	Expressions (supersedes 26.6.26 of IEEE Std 1364)	468
27.40	Atomic statement (supersedes atomic stmt in 26.6.27 of IEEE Std 1364)	470
27.41	Event statement (supersedes event stmt in 26.6.27 of IEEE Std 1364)	471
27.42	Process (supersedes process in 26.6.27 of IEEE Std 1364)	471
27.43	Assignment (supersedes 26.6.28 of IEEE Std 1364)	471
27.44	Event control (supersedes 26.6.30 of IEEE Std 1364)	472
27.45	Waits (supersedes wait in 26.6.32 of IEEE Std 1364)	472
27.46	If, if-else (supersedes 26.6.35 of IEEE Std 1364)	472
27.47	Case, pattern (supersedes 26.6.36 of IEEE Std 1364)	473
27.48	Expect	473
27.49	For (supersedes 26.6.33 of IEEE Std 1364)	474
27.50	Do-while, foreach	474
27.51	Alias statement	475
27.52	Disables (supersedes 26.6.38 of IEEE Std 1364)	475
27.53	Return statement	475
27.54	Attribute (supersedes 26.6.42 of IEEE Std 1364)	476
27.55	Generates (supersedes 26.6.44 of IEEE Std 1364)	477
28.	SystemVerilog assertion API	480
28.1	Requirements	480
28.2	Static information	480
28.3	Dynamic information	481

28.4	Control functions	484
29.	SystemVerilog code coverage control and API	487
29.1	Requirements	487
29.2	SystemVerilog real-time coverage access	487
29.3	FSM recognition	491
29.4	VPI coverage extensions.....	494
30.	SystemVerilog data read API	498
30.1	Introduction.....	498
30.2	Requirements	498
30.3	Extensions to VPI enumerations.....	499
30.4	VPI object type additions.....	500
30.5	Object model diagrams	502
30.6	Usage extensions to VPI routines	502
30.7	VPI routines added in SystemVerilog	505
30.8	Reading data	506
30.9	Optionally unloading data.....	506
30.10	Reading data from multiple databases and/or different read library providers	506
30.11	VPI routines extended in SystemVerilog.....	519
30.12	VPI routines added in SystemVerilog	520
	Annex A (normative) Formal syntax	524
	Annex B (normative) Keywords.....	564
	Annex C (normative) Std package.....	566
	Annex D (normative) Linked lists	568
	Annex E (normative) Formal semantics of concurrent assertions	576
	Annex F (normative) DPI C layer.....	578
	Annex G (normative) Include file svdpi.h	620
	Annex H (normative) Inclusion of foreign language code	630
	Annex I (normative) sv_vpi_user.h	634
	Annex J (informative) Glossary.....	644
	Annex K (informative) Bibliography	648
	Annex L (informative) List of participants.....	650
	Index	652

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**Standard For 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 Publication(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. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements 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.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC 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 transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC/IEEE 62530 has been processed through Technical Committee 93: Design automation.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1800(2005)	93/252/FDIS	93/263/RVD

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

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

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

IEC/IEEE Dual Logo International Standards

This Dual Logo International Standard is the result of an agreement between the IEC and the Institute of Electrical and Electronics Engineers, Inc. (IEEE). The original IEEE Standard was submitted to the IEC for consideration under the agreement, and the resulting IEC/IEEE Dual Logo International Standard has been published in accordance with the ISO/IEC Directives.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The 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 the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEC/IEEE Dual Logo International Standard is wholly voluntary. The IEC and IEEE disclaim liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEC or IEEE Standard document.

The IEC and IEEE do not warrant or represent the accuracy or content of the material contained herein, and expressly disclaim any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEC/IEEE Dual Logo International Standards documents are supplied "AS IS".

The existence of an IEC/IEEE Dual Logo International 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 IEC/IEEE Dual Logo International 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.

Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, 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 publishing and making this document available, the IEC and IEEE are not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Neither the IEC nor IEEE is undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEC/IEEE Dual Logo International Standards or IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations – Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEC/IEEE Dual Logo International Standards are welcome from any interested party, regardless of membership affiliation with the IEC or IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA and/or General Secretary, IEC, 3, rue de Varembe, PO Box 131, 1211 Geneva 20, Switzerland.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, 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.

NOTE – 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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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 20 March 2006

American National Standards Institute

Approved 8 November 2005

IEEE-SA Standards Board

Abstract: This standard provides a set of extensions to the IEEE 1364™ Verilog® hardware description language (HDL) to aid in the creation and verification of abstract architectural level models. It also includes design specification methods, embedded assertions language, testbench language including coverage and an assertions application programming interface (API), and a direct programming interface (DPI). This standard enables a productivity boost in design and validation and covers design, simulation, validation, and formal assertion-based verification flows.

Keywords: assertions, design automation, design verification, hardware description language, HDL, PLI, programming language interface, SystemVerilog, Verilog, Verilog programming interface, VPI

IEEE Introduction

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 languages (HDLs) presently used by designers while providing the capabilities lacking in those languages.

SystemVerilog is a unified hardware design, specification, and verification language that is based on the Accellera SystemVerilog 3.1a extensions to the Verilog HDL [B1]^a, 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 that is based on manual or automatic methodologies. SystemVerilog offers application programming interfaces (APIs) for coverage and assertions, a vendor-independent API to access proprietary waveform file formats, 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. This standardization project will provide the VLSI design engineers with a well-defined IEEE standard that meets their requirements in design and validation and 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 which they can support in order to deliver their solutions in this area.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

^aThe numbers in brackets correspond to the numbers in the bibliography in [Annex K](#).

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

1. Overview

1.1 Scope

This standard specifies extensions for a higher level of abstraction for modeling and verification with the Verilog® hardware description language (HDL). These additions extend Verilog into the systems space and the verification space. SystemVerilog is built on top of IEEE Std 1364™¹ for the Verilog HDL. This standard includes design specification methods, embedded assertions language, testbench language including coverage and assertions application programming interface (API), and a direct programming interface (DPI).

Throughout this standard, the following terms apply:

- *Verilog* refers to IEEE Std 1364 for the Verilog HDL.
- *Verilog-2001* refers to IEEE Std 1364-2001 [B4]² for the Verilog HDL.
- *Verilog-1995* refers to IEEE Std 1364-1995 [B3] for the Verilog HDL.
- *SystemVerilog* refers to the extensions to the Verilog standard (IEEE Std 1364) as defined in this standard.

2. Normative references

The following referenced documents are indispensable for the application of this standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1364™, IEEE Standard for Verilog Hardware Description Language.^{3, 4, 5}

IEEE Std 754™, IEEE Standard for Binary Floating-Point Arithmetic.

³IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org/>).

⁴This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

⁵The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.