



IEEE Standard for Verilog[®] Hardware Description Language

IEEE Computer Society

Sponsored by the
Design Automation Standards Committee

1364TM

IEEE
3 Park Avenue
New York, NY 10016-5997, USA
7 April 2006

IEEE Std 1364TM-2005
(Revision of IEEE Std 1364-2001)

[This is a preview. Click here to purchase the full publication.](#)

IEEE Standard for Verilog® Hardware Description Language

Sponsor

Design Automation Standards Committee
of the
IEEE Computer Society

Abstract: The Verilog hardware description language (HDL) is defined in this standard. Verilog HDL is a formal notation intended for use in all phases of the creation of electronic systems. Because it is both machine-readable and human-readable, it supports the development, verification, synthesis, and testing of hardware designs; the communication of hardware design data; and the maintenance, modification, and procurement of hardware. The primary audiences for this standard are the implementors of tools supporting the language and advanced users of the language.

Keywords: computer, computer languages, digital systems, electronic systems, hardware, hardware description languages, hardware design, HDL, PLI, programming language interface, Verilog, Verilog HDL, Verilog PLI

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

Copyright © 2006 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 7 April 2006. Printed in the United States of America.

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

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

Print: ISBN 0-7381-4850-4 SH95395
PDF: ISBN 0-7381-4851-2 SS95395

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.

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 IEEE Standard is wholly voluntary. The IEEE disclaims 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 IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims 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. IEEE Standards documents are supplied “**AS IS**.”

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. 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 IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other 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. 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, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with 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
Piscataway, NJ 08854
USA

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.

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.

Introduction

This introduction is not a part of IEEE Std 1364-2005, IEEE Standard for Verilog[®] Hardware Description Language.

The Verilog hardware description language (HDL) became an IEEE standard in 1995 as IEEE Std 1364-1995. It was designed to be simple, intuitive, and effective at multiple levels of abstraction in a standard textual format for a variety of design tools, including verification simulation, timing analysis, test analysis, and synthesis. It is because of these rich features that Verilog has been accepted to be the language of choice by an overwhelming number of integrated circuit (IC) designers.

Verilog contains a rich set of built-in primitives, including logic gates, user-definable primitives, switches, and wired logic. It also has device pin-to-pin delays and timing checks. The mixing of abstract levels is essentially provided by the semantics of two data types: nets and variables. Continuous assignments, in which expressions of both variables and nets can continuously drive values onto nets, provide the basic structural construct. Procedural assignments, in which the results of calculations involving variable and net values can be stored into variables, provide the basic behavioral construct. A design consists of a set of modules, each of which has an input/output (I/O) interface, and a description of its function, which can be structural, behavioral, or a mix. These modules are formed into a hierarchy and are interconnected with nets.

The Verilog language is extensible via the programming language interface (PLI) and the Verilog procedural interface (VPI) routines. The PLI/VPI is a collection of routines that allows foreign functions to access information contained in a Verilog HDL description of the design and facilitates dynamic interaction with simulation. Applications of PLI/VPI include connecting to a Verilog HDL simulator with other simulation and computer-assisted design (CAD) systems, customized debugging tasks, delay calculators, and annotators.

The language that influenced Verilog HDL the most was HILO-2, which was developed at Brunel University in England under a contract to produce a test generation system for the British Ministry of Defense. HILO-2 successfully combined the gate and register transfer levels of abstraction and supported verification simulation, timing analysis, fault simulation, and test generation.

In 1990, Cadence Design Systems placed the Verilog HDL into the public domain and the independent Open Verilog International (OVI) was formed to manage and promote Verilog HDL. In 1992, the Board of Directors of OVI began an effort to establish Verilog HDL as an IEEE standard. In 1993, the first IEEE working group was formed; and after 18 months of focused efforts, Verilog became an IEEE standard as IEEE Std 1364-1995.

After the standardization process was complete, the IEEE P1364 Working Group started looking for feedback from IEEE 1364 users worldwide so the standard could be enhanced and modified accordingly. This led to a five-year effort to get a much better Verilog standard in IEEE Std 1364-2001.

With the completion of IEEE Std 1364-2001, work continued in the larger Verilog community to identify outstanding issues with the language as well as ideas for possible enhancements. As Accellera began working on standardizing SystemVerilog in 2001, additional issues were identified that could possibly have led to incompatibilities between Verilog 1364 and SystemVerilog. The IEEE P1364 Working Group was established as a subcommittee of the SystemVerilog P1800 Working Group to help ensure consistent resolution of such issues. The result of this collaborative work is this standard, IEEE Std 1364-2005.

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.

Participants

At the time this standard was completed, the IEEE P1364 Working Group had the following membership:

Johny Srouji, IBM, *IEEE SystemVerilog Working Group Chair*
Tom Fitzpatrick, Mentor Graphics Corporation, *Chair*
Neil Korpusik, Sun Microsystems, Inc., *Co-chair*
Stuart Sutherland, Sutherland HDL, Inc., *Editor*
Shalom Bresticker, Intel Corporation, *Editor through February 2005*

The Errata Task Force had the following membership:

Karen Pieper, Synopsys, Inc., *Chair*

Kurt Baty, WFSDB Consulting
Stefen Boyd, Boyd Technology
Shalom Bresticker, Intel Corporation
Dennis Brophy, Mentor Graphics Corporation
Cliff Cummings, Sunburst Design, Inc.
Charles Dawson, Cadence Design Systems, Inc.
Tom Fitzpatrick, Mentor Graphics Corporation
Ronald Goodstein, First Shot Logic Simulation and Design
Mark Hartoog, Synopsys, Inc.
James Markevitch, Evergreen Technology Group

Dennis Marsa, Xilinx
Francoise Martinolle, Cadence Design Systems, Inc.
Mike McNamara, Verisity, Ltd.
Don Mills, LCDM Engineering
Anders Nordstrom, Cadence Design Systems, Inc.
Karen Pieper, Synopsys, Inc.
Brad Pierce, Synopsys, Inc.
Steven Sharp, Cadence Design Systems, Inc.
Alec Stanculescu, Fintronic USA, Inc.
Stuart Sutherland, Sutherland HDL, Inc.
Gordon Vreugdenhil, Mentor Graphics Corporation
Jason Woolf, Cadence Design Systems, Inc.

The Behavioral Task Force had the following membership:

Steven Sharp, Cadence Design Systems, Inc., *Chair*

Kurt Baty, WFSDB Consulting
Stefen Boyd, Boyd Technology
Shalom Bresticker, Intel Corporation
Dennis Brophy, Mentor Graphics Corporation
Cliff Cummings, Sunburst Design, Inc.
Steven Dovich, Cadence Design Systems, Inc.
Tom Fitzpatrick, Mentor Graphics Corporation
Ronald Goodstein, First Shot Logic Simulation and Design
Keith Gover, Mentor Graphics Corporation
Mark Hartoog, Synopsys, Inc.
Ennis Hawk, Jeda Technologies
Atsushi Kasuya, Jeda Technologies

Jay Lawrence, Cadence Design Systems, Inc.
Francoise Martinolle, Cadence Design Systems, Inc.
Kathryn McKinley, Cadence Design Systems, Inc.
Michael McNamara, Verisity, Ltd.
Don Mills, LCDM Engineering
Mehdi Mohtashemi, Synopsys, Inc.
Karen Pieper, Synopsys, Inc.
Brad Pierce, Synopsys, Inc.
Dave Rich, Mentor Graphics Corporation
Steven Sharp, Cadence Design Systems, Inc.
Alec Stanculescu, Fintronic, USA
Stuart Sutherland, Sutherland HDL, Inc.
Gordon Vreugdenhil, Mentor Graphics Corporation

The PLI Task Force had the following membership:

Charles Dawson, Cadence Design Systems, Inc., *Chair*
Ghassan Khoory, Synopsys, Inc., *Co-chair*

Tapati Basu, Synopsys, Inc.
Steven Dovich, Cadence Design Systems, Inc.
Ralph Duncan, Mentor Graphics Corporation
Jim Garnett, Mentor Graphics Corporation
Joao Geada, CLK Design Automation
Andrzej Litwiniuk, Synopsys, Inc.
Francoise Martinolle, Cadence Design Systems, Inc.
Sachchidananda Patel, Synopsys, Inc.

Michael Rohleder, Freescale Semiconductor, Inc.
Rob Slater, Freescale Semiconductor, Inc.
John Stickley, Mentor Graphics Corporation
Stuart Sutherland, Sutherland HDL, Inc.
Bassam Tabbara, Novas Software, Inc.
Jim Vellenga, Cadence Design Systems, Inc.
Doug Warmke, Mentor Graphics Corporation

In addition, the working group wishes to recognize the substantial efforts of past contributors:

Michael McNamara, Cadence Design Systems, Inc.,
1364 Working Group past chair (through September 2004)
Alec Stanculescu, Fintronic USA, *1364 Working Group past vice-chair (through June 2004)*
Stefen Boyd, Boyd Technology, *ETF past co-chair (through November 2004)*

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

Accellera
Bluespec, Inc.
Cadence Design Systems, Inc.
Fintronic U.S.A.
IBM
Infineon Technologies

Intel Corporation
Mentor Graphics Corporation
Sun Microsystems, Inc.
Sunburst Design, Inc.
Sutherland HDL, Inc.
Synopsys, Inc.

When the IEEE-SA Standards Board approved this standard on 8 November 2005, it had the following membership:

Steve M. Mills, *Chair*
Richard H. Hulett, *Vice Chair*
Don Wright, *Past Chair*
Judith Gorman, *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman

William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolic

T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan H. Cookson, *NIST Representative*

Michelle D. Turner
IEEE Standards Project Editor

Contents

1.	Overview	1
1.1	Scope	1
1.2	Conventions used in this standard	1
1.3	Syntactic description	2
1.4	Use of color in this standard	3
1.5	Contents of this standard	3
1.6	Deprecated clauses	5
1.7	Header file listings	5
1.8	Examples	5
1.9	Prerequisites	5
2.	Normative references	6
3.	Lexical conventions	8
3.1	Lexical tokens	8
3.2	White space	8
3.3	Comments	8
3.4	Operators	8
3.5	Numbers	9
3.5.1	Integer constants	10
3.5.2	Real constants	12
3.5.3	Conversion	12
3.6	Strings	12
3.6.1	String variable declaration	13
3.6.2	String manipulation	13
3.6.3	Special characters in strings	13
3.7	Identifiers, keywords, and system names	14
3.7.1	Escaped identifiers	14
3.7.2	Keywords	15
3.7.3	System tasks and functions	15
3.7.4	Compiler directives	15
3.8	Attributes	16
3.8.1	Examples	16
3.8.2	Syntax	18
4.	Data types	21
4.1	Value set	21
4.2	Nets and variables	21
4.2.1	Net declarations	21
4.2.2	Variable declarations	23
4.3	Vectors	24
4.3.1	Specifying vectors	24
4.3.2	Vector net accessibility	24
4.4	Strengths	25
4.4.1	Charge strength	25
4.4.2	Drive strength	25
4.5	Implicit declarations	25
4.6	Net types	26
4.6.1	Wire and tri nets	26
4.6.2	Wired nets	27
4.6.3	Trireg net	28

4.6.4	Tri0 and tri1 nets	31
4.6.5	Unresolved nets	31
4.6.6	Supply nets	32
4.7	Regs	32
4.8	Integers, reals, times, and reals	32
4.8.1	Operators and real numbers	33
4.8.2	Conversion	33
4.9	Arrays	34
4.9.1	Net arrays	34
4.9.2	reg and variable arrays	34
4.9.3	Memories	35
4.10	Parameters	35
4.10.1	Module parameters	36
4.10.2	Local parameters (localparam)	37
4.10.3	Specify parameters	38
4.11	Name spaces	39
5.	Expressions	41
5.1	Operators	41
5.1.1	Operators with real operands	42
5.1.2	Operator precedence	43
5.1.3	Using integer numbers in expressions	44
5.1.4	Expression evaluation order	45
5.1.5	Arithmetic operators	45
5.1.6	Arithmetic expressions with regs and integers	47
5.1.7	Relational operators	48
5.1.8	Equality operators	49
5.1.9	Logical operators	49
5.1.10	Bitwise operators	50
5.1.11	Reduction operators	51
5.1.12	Shift operators	53
5.1.13	Conditional operator	53
5.1.14	Concatenations	54
5.2	Operands	55
5.2.1	Vector bit-select and part-select addressing	56
5.2.2	Array and memory addressing	57
5.2.3	Strings	58
5.3	Minimum, typical, and maximum delay expressions	61
5.4	Expression bit lengths	62
5.4.1	Rules for expression bit lengths	62
5.4.2	Example of expression bit-length problem	63
5.4.3	Example of self-determined expressions	64
5.5	Signed expressions	64
5.5.1	Rules for expression types	65
5.5.2	Steps for evaluating an expression	65
5.5.3	Steps for evaluating an assignment	66
5.5.4	Handling X and Z in signed expressions	66
5.6	Assignments and truncation	66
6.	Assignments	68
6.1	Continuous assignments	68
6.1.1	The net declaration assignment	69
6.1.2	The continuous assignment statement	69
6.1.3	Delays	71

6.1.4	Strength	71
6.2	Procedural assignments	72
6.2.1	Variable declaration assignment	72
6.2.2	Variable declaration syntax	73
7.	Gate- and switch-level modeling	74
7.1	Gate and switch declaration syntax	74
7.1.1	The gate type specification	76
7.1.2	The drive strength specification	76
7.1.3	The delay specification	77
7.1.4	The primitive instance identifier	77
7.1.5	The range specification	77
7.1.6	Primitive instance connection list	78
7.2	and, nand, nor, or, xor, and xnor gates	80
7.3	buf and not gates	81
7.4	bufif1, bufif0, notif1, and notif0 gates	82
7.5	MOS switches	83
7.6	Bidirectional pass switches	84
7.7	CMOS switches	85
7.8	pullup and pulldown sources	86
7.9	Logic strength modeling	86
7.10	Strengths and values of combined signals	88
7.10.1	Combined signals of unambiguous strength	88
7.10.2	Ambiguous strengths: sources and combinations	89
7.10.3	Ambiguous strength signals and unambiguous signals	94
7.10.4	Wired logic net types	98
7.11	Strength reduction by nonresistive devices	100
7.12	Strength reduction by resistive devices	100
7.13	Strengths of net types	100
7.13.1	tri0 and tri1 net strengths	100
7.13.2	trireg strength	100
7.13.3	supply0 and supply1 net strengths	101
7.14	Gate and net delays	101
7.14.1	min:typ:max delays	102
7.14.2	trireg net charge decay	103
8.	User-defined primitives (UDPs)	105
8.1	UDP definition	105
8.1.1	UDP header	107
8.1.2	UDP port declarations	107
8.1.3	Sequential UDP initial statement	107
8.1.4	UDP state table	107
8.1.5	Z values in UDP	108
8.1.6	Summary of symbols	108
8.2	Combinational UDPs	109
8.3	Level-sensitive sequential UDPs	110
8.4	Edge-sensitive sequential UDPs	110
8.5	Sequential UDP initialization	111
8.6	UDP instances	113
8.7	Mixing level-sensitive and edge-sensitive descriptions	114
8.8	Level-sensitive dominance	115
9.	Behavioral modeling	116
9.1	Behavioral model overview	116