INTERNATIONAL STANDARD

IEC 61691-4

First edition
2004-10

IEEE 1364™

Behavioural languages –

Part 4:
Verilog® hardware description language

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

The Institute of Electrical and Electronics Engineers, Inc, 3 Park Avenue, New York, NY 10016-5997, USA
Telephone: +1 732 562 3800 Telefax: +1 732 562 1571 E-mail: stds-info@ieee.org Web: www.standards.ieee.org

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

BEHAVIOURAL LANGUAGES –

Part 4: Verilog® hardware description 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 61691-4 has been processed through IEC technical committee 93: Design automation.

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

IEEE Std	FDIS	Report on voting
1364 (2001)	93/192/FDIS	93/197/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.

This publication has been drafted in accordance with the ISO/IEC Directives.

The committee has decided that the contents of this publication will remain unchanged until 2006.

IEC 61691 consists of the following parts, under the general title *Behavioural languages:*

IEC/IEEE 61691-1-1, Part 1: *VHDL language reference manual*

IEC 61691-2, Part 2: *VHDL multilogic system for model interoperability*

IEC 61691-3-1, Part 3-1: *Analog description in VHDL* (under consideration)

IEC 61691-3-2, Part 3-2: *Mathematical operation in VHDL*

IEC 61691-3-3, Part 3-3: *Synthesis in VHDL*

IEC 61691-3-4, Part 3-4: *Timing expressions in VHDL* (under consideration)

IEC 61691-3-5, Part 3-5: *Library utilities in VHDL* (under consideration)

IEC/IEEE 61691-4, Part 4: *Verilog® hardware description language*

IEC/IEEE 61691-5, Part 5: *VITAL ASIC (application specific integrated circuit) modeling specification*

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 Varembé, 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.

IEEE Standard Verilog® Hardware Description Language

Sponsor

Design Automation Standards Committee
of the
IEEE Computer Society

Approved 17 March 2001

IEEE-SA Standards Board

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 HDL, Verilog PLI, Verilog®

IEEE Introduction

The Verilog®* Hardware Description Language (Verilog 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 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 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 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 1364 Working Group started looking for feedback from 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.

Objective of the IEEE Std 1364-2001 effort

The starting point for the IEEE 1364 Working Group for this standard was the feedback received from the IEEE Std 1364-1995 users worldwide. It was clear from the feedback that users wanted improvements in all aspects of the language. Users at the higher levels wanted to expand and improve the language at the RTL and behavioral levels, while users at the lower levels wanted improved capability for ASIC designs and signoff. It was for this reason that the 1364 Working Group was organized into three task forces: Behavioral, ASIC, and PLI.

---

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

The clear directive from the users for these three task forces was to start by solving some of the following problems:

Consolidate existing IEEE Std 1364-1995

Verilog Generate statement

Multi-dimensional arrays

Enhanced Verilog file I/O

Re-entrant tasks

Standardize Verilog configurations

Enhance timing representation

Enhance the VPI routines

Achievements

Over a period of four years the 1364 Verilog Standards Group (VSG) has produced five drafts of the LRM. The three task forces went through the IEEE Std 1364-1995 LRM very thoroughly and in the process of consolidating the existing LRM have been able to provide nearly three hundred clarifications and errata for the Behavioral, ASIC, and PLI sections. In addition, the VSG has also been able to agree on all the enhancements that were requested (including the ones stated above).

Three new sections have been added. Clause 13, Configuring the contents of a design, deals with configuration management and has been added to facilitate both the sharing of Verilog designs between designers and/or design groups and the repeatability of the exact contents of a given simulation session. Clause 15, Timing checks, has been broken out of Clause 17, System tasks and functions, and details more fully how timing checks are used in specify blocks. Clause 16, Backannotation using the Standard Delay Format (SDF), addresses using back annotation (IEEE Std 1497-1999) within IEEE Std 1364-2001.

Extreme care has been taken to enhance the VPI routines to handle all the enhancements in the Behavioral and other areas of the LRM. Minimum work has been done on the PLI routines and most of the work has been concentrated on the VPI routines. Some of the enhancements in the VPI are the save and restart, simulation control, work area access, error handling, assign/deassign and support for array of instances, generate, and file I/O.

Work on this standard would not have been possible without funding from the CAS society of the IEEE and Open Verilog International.

The IEEE Std 1364-2001 Verilog Standards Group organization

Many individuals from many different organizations participated directly or indirectly in the standardization process. The main body of the IEEE Std 1364-2001 working group is located in the United States, with a subgroup in Japan (EIAJ/1364HDL).

The members of the IEEE Std 1364-2001 working group had voting privileges and all motions had to be approved by this group to be implemented. The three task forces focused on their specific areas and their recommendations were eventually voted on by the IEEE Std 1364-2001 working group.

BEHAVIOURAL LANGUAGES –

Part 4: Verilog® hardware description language

1. Overview

1.1 Objectives of this standard

The intent of this standard is to serve as a complete specification of the Verilog® Hardware Description Language (HDL). This document contains

- The formal syntax and semantics of all Verilog HDL constructs
- The formal syntax and semantics of Standard Delay Format (SDF) constructs
- Simulation system tasks and functions, such as text output display commands
- Compiler directives, such as text substitution macros and simulation time scaling
- The Programming Language Interface (PLI) binding mechanism
- The formal syntax and semantics of access routines, task/function routines, and Verilog procedural interface routines
- Informative usage examples
- Informative delay model for SDF
- Listings of header files for PLI

1.2 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 some examples and notes.

The term *shall* is used through out this standard to indicate mandatory requirements, whereas the term *can* is used to indicate optional features. These terms denote different meanings to different readers of this standard:

a) To the developers of tools that process the Verilog HDL, the term *shall* denotes a requirement that the standard imposes. The resulting implementation is required to enforce the requirements and to issue an error if the requirement is not met by the input.
b) To the Verilog HDL model developer, the term *shall* denotes that the characteristics of the Verilog HDL are natural consequences of the language definition. The model developer is required to adhere to the constraint implied by the characteristic. The term *can* denotes optional features that the model

developer can exercise at discretion. If used, however, the model developer is required to follow the requirements set forth by the language definition.

c) To the Verilog HDL model user, the term *shall* denotes that the characteristics of the models are natural consequences of the language definition. The model user can depend on the characteristics of the model implied by its Verilog HDL source text.

1.3 Syntactic description

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

a) Lowercase words, some containing embedded underscores, are used to denote syntactic categories. For example:

module_declaration

b) Boldface words are used to denote reserved keywords, operators, and punctuation marks as a required part of the syntax. These words appear in a larger font for distinction. For example:

module **=>** **;**

c) A vertical bar separates alternative items unless it appears in boldface, in which case it stands for itself. For example:

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

d) Square brackets enclose optional items. For example:

input_declaration ::= **input** [range] list_of_variables **;**

e) Braces enclose a repeated item unless it appears in boldface, in which case it stands for itself. 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_assignment **,** param_assignment

f) If the name of any category starts with an italicized part, it is equivalent to the category name without the italicized part. The italicized part is intended to convey some semantic information. For example, *msb*_constant_expression and *lsb*_constant_expression are equivalent to constant_expression.

The main text uses *italicized* font when a term is being defined, and `constant-width` font for examples, file names, and while referring to constants, especially `0`, `1`, `x`, and `z` values.

1.4 Contents of this standard

A synopsis of the clauses and annexes is presented as a quick reference. There are 27 clauses and 8 annexes. All clauses, as well as Annex A, Annex B, Annex E, Annex F, and Annex G, are normative parts of this standard. Annex C, Annex D, and Annex H are included for informative purposes only.

Clause 1—Overview: This clause discusses the conventions used in this standard and its contents.

Clause 2—This clause describes the lexical tokens used in Verilog HDL source text and their conventions.: This clause describes how to specify and interpret the lexical tokens.

Clause 3—Data types: This clause describes net and variable data types. This clause also discusses the parameter data type for constant values and describes drive and charge strength of the values on nets.

Clause 4—Expressions: This clause describes the operators and operands that can be used in expressions.

Clause 5—Scheduling semantics: This clause describes the scheduling semantics of the Verilog HDL.

Clause 6—Assignments: This clause compares the two main types of assignment statements in the Verilog HDL—continuous assignments and procedural assignments. It describes the continuous assignment statement that drives values onto nets.

Clause 7—Gate and switch level modeling: This clause describes the gate and switch level primitives and logic strength modeling.

Clause 8—User-defined primitives (UDPs): This clause describes how a primitive can be defined in the Verilog HDL and how these primitives are included in Verilog HDL models.

Clause 9—Behavioral modeling: This clause describes procedural assignments, procedural continuous assignments, and behavioral language statements.

Clause 10—Tasks and functions: This clause describes tasks and functions—procedures that can be called from more than one place in a behavioral model. It describes how tasks can be used like subroutines and how functions can be used to define new operators.

Clause 11—Disabling of named blocks and tasks: This clause describes how to disable the execution of a task and a block of statements that has a specified name.

Clause 12—Hierarchical structures: This clause describes how hierarchies are created in the Verilog HDL and how parameter values declared in a module can be overridden. It describes how generated instantiations can be used to do conditional or multiple instantiations in a design.

Clause 13—Configuring the contents of a design: This clause describes how to configure the contents of a design.

Clause 14—Specify blocks: This clause describes how to specify timing relationships between input and output ports of a module.

Clause 15—Timing checks: This clause describes how timing checks are used in specify blocks to determine if signals obey the timing constraints.

Clause 16—Backannotation using the Standard Delay Format (SDF): This clause describes syntax and semantics of Standard Delay Format (SDF) constructs.

Clause 17—System tasks and functions: This clause describes the system tasks and functions.

Clause 18—Value change dump (VCD) files: This clause describes the system tasks associated with Value Change Dump (VCD) file, and the format of the file.

Clause 19—Compiler directives: This clause describes the compiler directives.

Clause 20—PLI overview: This clause previews the C language procedural interface standard (Programming Language Interface or PLI) and interface mechanisms that are part of the Verilog HDL.

Clause 21—PLI TF and ACC interface mechanism
This clause describes the interface mechanism that provides a means for users to link PLI task/function (TF) routine and access (ACC) routine applications to Verilog software tools.

Clause 22—Using ACC routines: This clause describes the ACC routines in general, including how and why to use them.

Clause 23—ACC routine definitions: This clause describes the specific ACC routines, explaining their function, syntax, and usage.

Clause 24—Using TF routines: This clause provides an overview of the types of operations that are done with the TF routines.

Clause 25—TF routine definitions: This clause describes the specific TF routines, explaining their function, syntax, and usage.

Clause 26—Using VPI routines: This clause provides an overview of the types of operations that are done with the Verilog Programming Interface (VPI) routines.

Clause 27—VPI routine definitions: This clause describes the VPI routines.

Annex A—Formal syntax definition: This normative annex describes, using BNF, the syntax of the Verilog HDL.

Annex B—List of keywords: This normative annex lists the Verilog HDL keywords.

Annex C—System tasks and functions: This informative annex describes system tasks and functions that are frequently used, but that are not part of the standard.

Annex D—Compiler directives: This informative annex describes compiler directives that are frequently used, but that are not part of the standard.

Annex E—acc_user.h: This normative annex provides a listing of the contents of the `acc_user.h` file.

Annex F—veriuser.h: This normative annex provides a listing of the contents of the `vpi_user.h` file.

Annex G—vpi_user.h: This normative annex provides a listing of the contents of the `veriuser.h` file.

Annex H—Bibliography: This informative annex contains bibliographic entries pertaining to this standard.

1.5 Header file listings

The header file listings included in the annexes E, F, and G for `acc_user.h, veriuser.h`, and `vpi_user.h` are a normative part of this standard. All compliant software tools should use the same function declarations, constant definitions, and structure definitions contained in these header file listings.

1.6 Examples

Several small examples in the Verilog HDL and the C programming language are shown throughout this standard. These examples are ***informative***—they are intended to illustrate the usage of Verilog HDL constructs and PLI functions in a simple context and do not define the full syntax.

1.7 Prerequisites

Clauses 20 through 27 and Annexes E through G presuppose a working knowledge of the C programming language.