IEC 62531

Edition 2.0 2012-06

INTERNATIONAL STANDARD

IEEE Std 1850™

Property Specification Language (PSL)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XH

ICS 25.040; 35.060

ISBN 978-2-83220-106-0

Contents

Property Specification Language (PSL)

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.

   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, 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 http://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.

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.

International Standard IEC 62531/ IEEE Std 1850-2010 has been processed through IEC technical committee 93: Design automation, under the IEC/IEEE Dual Logo Agreement.

This second edition cancels and replaces the first edition, published in 2007, and constitutes a technical revision.

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

IEEE Std	FDIS	Report on voting
IEEE Std 1850-2010	93/319/FDIS	93/326/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 IEC Technical Committee and IEEE Technical Committee have decided that the contents of this publication will remain unchanged until the stability 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.

IEEE Std 1850™-2010
(Revision of
IEEE Std1850-2005)

IEEE Standard for Property Specification Language (PSL)

Sponsor

Design Automation Standards Committee
of the
IEEE Computer Society

and the
IEEE Standards Association Corporate Advisory Group

Approved 25 March 2010
IEEE-SA Standards Board

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

Accellera Property Specification Language Reference Manual (version 1.1), Accellera

GDL: General Description Language, Accellera, Mar. 2005

Abstract: The IEEE Property Specification Language (PSL) is defined. PSL is a formal notation for specification of electronic system behavior, compatible with multiple electronic system design languages, including IEEE Std 1076™ (VHDL®), IEEE Std 1354 (Verilog®), IEEE Std 1666™ (SystemC®), and IEEE Std 1800™ (SystemVerilog®), thereby enabling a common specification and verification flow for multi-language and mixed-language designs. PSL captures design intent in a form suitable for simulation, formal verification, formal analysis, and hybrid verification tools. PSL enhances communication among architects, designers, and verification engineers to increase productivity throughout the design and verification process. The primary audiences for this standard are the implementors of tools supporting the language and advanced users of the language.
Keywords: ABV, assertion, assertion-based verification, assumption, cover, model checking, property, PSL, specification, temporal logic, verification

IEEE Introduction

This introduction is not part of IEEE Std 1850-2010, IEEE Standard for Property Specification Language (PSL).

IEEE Std 1850 Property Specification Language (PSL) is based upon the Accellera Property Specification Language (Accellera PSL), a language for formal specification of electronic system behavior, which was developed by Accellera, a consortium of Electronic Design Automation (EDA), semiconductor, and system companies. IEEE Std 1850 PSL 2010 refines IEEE Std 1850 PSL 2005 by providing extensions for improved verification IP reuse (e.g., the vpkg type of vunit) and interaction between the assertions and the simulation environment (local variables), and by addressing minor technical issues. The formal semantics were updated to reflect these changes.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard 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

This document is copyrighted by the IEEE. It is made available 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 this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards 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. 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 Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

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 amendment 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 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 amendment 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.

Property Specification Language (PSL)

1. Overview

1.1 Scope

This standard defines the property specification language (PSL), which formally describes electronic system behavior. This standard specifies the syntax and semantics for PSL and also clarifies how PSL interfaces with various standard electronic system design languages.

1.2 Purpose

The purpose of this standard is to provide a well-defined language for formal specification of electronic system behavior, one that is compatible with multiple electronic system design languages, including IEEE Std 1076™ (VHDL®),[1] IEEE Std 1364™ (Verilog®), IEEE Std 1800™ (SystemVerilog®), and IEEE Std 1666™ (SystemC®), to facilitate a common specification and verification flow for multi-language and mixed-language designs.

This standard creates an updated IEEE standard based upon IEEE Std 1850-2005. The updated standard will refine IEEE standard, addressing errata, minor technical issues, and proposed extensions specifically related to property reuse and improved simulation usability.

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

1.2.1 Background

The complexity of Very Large Scale Integration (VLSI) has grown to such a degree that traditional approaches have begun to reach their limitations, and verification costs have reached 60%–70% of development resources. The need for advanced verification methodology, with improved observability of design behavior and improved controllability of the verification process, has become critical. Over the last decade, a methodology based on the notion of "properties" has been identified as a powerful verification paradigm that can assure enhanced productivity, higher design quality, and, ultimately, faster time to market and higher value to engineers and end-users of electronics products. Properties, as used in this context, are concise, declarative, expressive, and unambiguous specifications of desired system behavior that are used to guide the verification process. IEEE 1850 PSL is a standard language for specifying electronic system behavior using properties. PSL facilitates property-based verification using both simulation and formal verification, thereby enabling a productivity boost in functional verification.

1.2.2 Motivation

Ensuring that a design's implementation satisfies its specification is the foundation of hardware verification. Key to the design and verification process is the act of specification. Yet historically, the process of specification has consisted of creating a natural language description of a set of design requirements. This form of specification is both ambiguous and, in many cases, unverifiable due to the lack of a standard machine-executable representation. Furthermore, ensuring that all functional aspects of the specification have been adequately *verified* (that is, covered) is problematic.

The IEEE PSL was developed to address these shortcomings. It gives the design architect a standard means of specifying design properties using a concise syntax with clearly-defined formal semantics. Similarly, it enables the RTL implementer to capture design intent in a verifiable form, while enabling the verification engineer to validate that the implementation satisfies its specification through *dynamic* (that is, simulation) and *static* (that is, formal) verification means. Furthermore, it provides a means to measure the quality of the verification process through the creation of functional coverage models built on formally specified properties. In addition, it provides a standard means for hardware designers and verification engineers to create a rigorous and machine-executable design specification.

1.2.3 Goals

PSL was specifically developed to fulfill the following general hardware functional specification requirements:

— Easy to learn, write, and read
— Concise syntax
— Rigorously well-defined formal semantics
— Expressive power, permitting specifications of a large class of real-world design properties
— Known efficient underlying algorithms in simulation, as well as formal verification

1.3 Usage

PSL is a language for the formal specification of hardware. It is used to describe properties that are required to hold in the design under verification. PSL provides a means to write specifications that are both easy to read and mathematically precise. It is intended to be used for functional specification on the one hand and as input to functional verification tools on the other. Thus, a PSL specification is an executable specification of a hardware design.

1.3.1 Functional specification

PSL can be used to capture requirements regarding the overall behavior of a design, as well as assumptions about the environment in which the design is expected to operate. PSL can also capture internal behavioral requirements and assumptions that arise during the design process. Both enable more effective functional verification and reuse of the design.

One important use of PSL is for documentation, either in place of or along with an English specification. A PSL specification can describe simple invariants (for example, signals `read_enable` and `write_enable` are never asserted simultaneously) as well as multi-cycle behavior (for example, correct behavior of an interface with respect to a bus protocol or correct behavior of pipelined operations).

A PSL specification consists of *assertions* regarding *properties* of a design under a set of *assumptions*. A *property* is built from three kinds of elements: *Boolean expressions*, which describe behavior over one cycle; *sequential expressions*, which can describe multi-cycle behavior; and *temporal operators*, which describe temporal relationships among Boolean expressions and sequences. For example, consider the following Verilog Boolean expression:

```
ena || enb
```

This expression describes a cycle in which at least one of the signals `ena` and `enb` are asserted. The PSL sequential expression

```
{req; ack; !cancel}
```

describes a sequence of cycles, such that `req` is asserted in the first cycle, `ack` is asserted in the second cycle, and `cancel` is deasserted in the third cycle. The following property, obtained by applying the temporal operators `always` and `|=>` to these expressions,

```
always {req;ack;!cancel} |=> (ena || enb)
```

means that always (that is, in every cycle), if the sequence `{req;ack;!cancel}` occurs, then either `ena` or `enb` is asserted one cycle after the sequence ends. Adding the directive `assert` as follows:

```
assert always {req;ack;!cancel} |=> (ena || enb);
```

completes the specification, indicating that this property is expected to hold in the design and that this expectation needs to be verified.

1.3.2 Functional verification

PSL can also be used as input to verification tools, for both verification by simulation, as well as formal verification using a model checker or a theorem prover. Each of these is discussed in the subclauses that follow.

1.3.2.1 Simulation

A PSL specification can also be used to automatically generate checks of simulated behavior. This can be done, for example, by directly integrating the checks in the simulation tool; by interpreting PSL properties in a testbench automation tool that drives the simulator; by generating HDL monitors that are simulated alongside the design; or by analyzing the traces produced during simulation.

For instance, the following PSL property:

Property 1: `always (req -> next !req)`

states that signal `req` is a pulsed signal, i.e., if it is high in some cycle, then it is low in the following cycle. Such a property can be easily checked using a simulation checker written in some HDL that has the functionality of the finite state machine (FSM) shown in Figure 1.

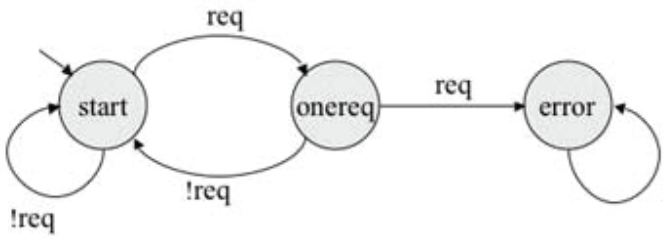

Figure 1—A simple (deterministic) FSM that checks Property 1

For properties more complicated than the property shown in Figure 1, manually writing a corresponding checker is painstaking and error-prone, and maintaining a collection of such checkers for a constantly changing design under development is a time-consuming task. Instead, a PSL specification can be used as input to a tool that automatically generates simulatable checkers.

Although in principle, all PSL properties can be checked for finite paths in simulation, the implementation of the checks is often significantly simpler for a subset called the *simple subset* of PSL. Informally, in this subset, composition of temporal properties is restricted to ensure that time *moves forward* from left to right through a property, as it does in a timing diagram. (See 4.4.4 for the formal definition of the simple subset.) For example, the property

Property 2: `always (a -> next[3] b)`

which states that, if `a` is asserted, then `b` is asserted three cycles later, belongs to the simple subset, because `a` appears to the left of `b` in the property and also appears to the left of `b` in the timing diagram of any behavior that is not a violation of the property. Figure 2 shows an example of such a timing diagram.

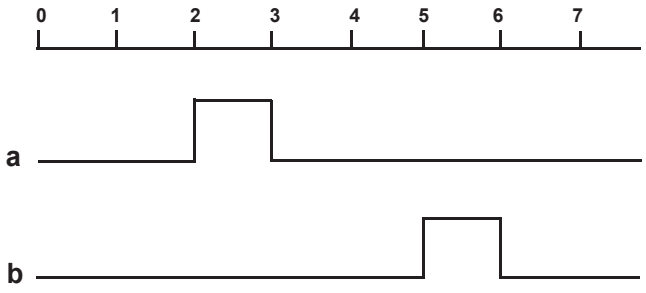

Figure 2—A trace that satisfies Property 2

An example of a property that is not in this subset is the property

Property 3: `always ((a && next[3] b) -> c)`

which states that, if `a` is asserted and `b` is asserted three cycles later, then `c` is asserted (in the same cycle as `a`). This property does not belong to the simple subset, because although `c` appears to the right of `a` and `b` in

the property, it appears to the left of `b` in a timing diagram that is not a violation of the property. Figure 3 shows an example of such a timing diagram.

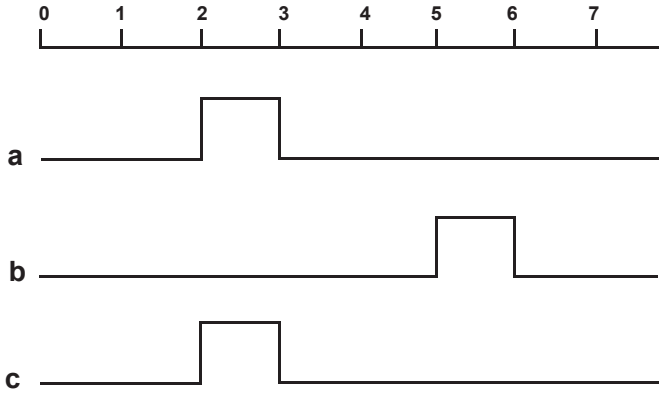

Figure 3—A trace that satisfies Property 3

1.3.2.2 Formal verification

PSL is an extension of the standard temporal logics Linear-Time Temporal Logic (LTL) and Computation Tree Logic (CTL). A specification in the PSL Foundation Language (respectively, the PSL Optional Branching Extension) can be *compiled down* to a formula of pure LTL (respectively, CTL), possibly with some auxiliary HDL code, known as a *satellite*.

2. Normative references

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

"General Description Language," Accellera, Napa, CA, Mar. 2005.[2]

IEC/IEEE 62142 (IEEE Std 1364.1), Standard for Verilog Register Transfer Level Synthesis.[3]

IEEE Std 1076™, IEEE Standard VHDL Language Reference Manual.[4, 5]

IEEE Std 1076.6™, IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis.

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

IEEE Std 1666™, IEEE Standard for the SystemC Language.

IEEE Std 1800™, IEEE Standard for the SystemVerilog Language.

[2]This document is available from the IEEE Standards World Wide Web site, at http://standards.ieee.org/downloads/1850/1850-2005/gdl.pdf.

[3]IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3, rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iec.ch/). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[4]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

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