FP6-IST-507219

PROSYD:

Property-Based System Design

Instrument: Specific Targeted Research Project

Thematic Priority: Information Society Technologies

Annual Language Study and Recommendation to Accellera (Deliverable 4.3/1)

Due date of deliverable: December 31, 2004
Actual submission date: March 29, 2005

Start date of project: January 1, 2004 Duration: Three years

Organisation name of lead contractor for this deliverable: IBM

Revision 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	☒
PP	Restricted to other programme participants (including the Commission Services)	☐
RE	Restricted to a group specified by the consortium (including the Commission Services)	☐
CO	Confidential, only for members of the consortium (including the Commission Services)	☐

Notices

For information, contact eisner@il.ibm.com.

This document is intended to fulfil the contractual obligations of the PROSYD project concerning deliverable 4.3/1 described in contract number 507219.

Table of Revisions

Version	Date	Description and Reason	By	Affected Sections
0.1	14.3.05	First draft by authors	Cindy Eisner	All
0.2	22.3.05	Update after review by partners	Cindy Eisner	2
1.0	29.3.05	Final version	Cindy Eisner	Version number, Executive Summary

Authors

Lyes Benalycherif

Cindy Eisner

Andrea Fedeli

Dana Fisman

Anthony McIsaac

Adriana Maggiore

Avigail Orni

Christian Pichler

Simone Semprini

Klaus Winkelmann

Executive Summary

This document contains the 2004 combined PSL language recommendations of all partners of the PROSYD consortium to Accellera. We record various proposals we have made to Accellera as a result of our work on PROSYD in 2004, make a number of new recommendations, and give a preview of an issue that we will be considering in the following years.

Purpose

The purpose of this document is to distil our experience of the past year with PSL into concrete recommendations as to language extensions.

Intended Audience

This report is intended for Mr. Harry Foster (a consortium member), in his role as the Chairman of the Accellera Formal Verification Technical Committee, and in his follow-on role as Chairman of IEEE P1850, which is following Accellera's recommendation to further standardize Accellera PSL as IEEE-1850, and for the members of the IEEE P1850 working group.

Background

The consortium began working with PSL on January 1, 2004, as part of the PROSYD project (see www.prosyd.org). During the course of the last year and more, we have had numerous opportunities to experience firsthand the advantages of the language as well as areas for improvement. In this document, we summarize our recommendations to Accellera based on that experience.

Contents

Glossary

Flavour

PSL comes in various flavours, corresponding to the main hardware description languages. Two of the flavours and Verilog and VHDL.

HDL

Hardware description language.

IEEE

Institute of Electrical and Electronics Engineers. A non-profit, technical professional association of more than 360,000 individual members in approximately 175 countries.

PSL

Property Specification Language, the language for specification of designs upon which PROSYD is based.

SERE

Sequential Extended Regular Expression, a basic type of temporal expression in PSL.

Sequence

A specific type of SERE.

Structural contradiction

A contradiction which happens because of incompatible temporal requirements (e.g. that a SERE last for N cycles and also for N+1 cycles), as opposed to a logical contradiction that happens because of incompatible logical requirements (e.g. that both "b" and "not b" hold simultaneously).

Suffix implication

A type of temporal property in PSL.

Verilog

One of two standardized HDLs used to specify the structure and behaviour of electronic systems in textual format. Developed in the mid-1980s as a proprietary language and acquired by Cadence Design Systems, it became a de facto industry standard. In the mid-90s Cadence placed it into the public domain and it became a de jure standard promulgated by the IEEE. Verilog is also the name of a legacy simulation tool offered by Cadence.

VHDL

The second of two standardized HDLs used to specify the structure and behaviour of electronic systems in textual format. VHDL stands for VHSIC Hardware Description Language. It was developed starting in 1983 with support the U.S. government, and in 1987 was adopted by the IEEE as a standard (IEEE 1076).

1 Introduction

During the past year, the consortium has had much opportunity to gain experience with the language, both from the point of view of users and from that of tool development. This report summarizes our recommendations based upon that experience.

There are three relevant versions of the standard:

- Accellera PSL Version 1.01 was the official version as of the start of this project. It was released on April 25, 2003 and approved by Accellera on May 29, 2003.
- Accellera PSL Version 1.1 was released on June 9, 2004 and was approved by Accellera on June 9, 2004.
- The first IEEE version of PSL is now undergoing the approval process.

Some of our recommendations were given early in 2004 by email and were already incorporated into Accellera Version 1.1. Some were given by email after June 2004 and have been considered already for the first IEEE version. Others are completely new and are being given in this report for the first time. Finally, we raise some subjects for which we do not yet have a firm recommendation. Rather, we raise them here in order to make the Accellera and IEEE committees aware of the general areas in which we are thinking, and in which we may make more concrete recommendations in the coming years.

The recommendations are detailed in the following section.

2 Recommendations

In this section we detail the recommendations. For recommendations that have already been incorporated into Accellera Version 1.1 or into the draft of the first IEEE version, we outline the recommendation as it was previously made, and its current status. For other recommendations, we give either a concrete proposal or a general outline of the direction of our thoughts, depending on our degree of progress in the understanding of the issue.

Previously submitted recommendations

For the record, we summarize here issues that were previously submitted to Accellera (in a number of emails throughout 2004) and were already considered by Accellera in the drafting of Accellera Version 1.1, or by the IEEE committee in the drafting of the first IEEE version.

1. HDL Embedding: The syntax {a} is ambiguous in the Verilog flavour. It can be either a SERE consisting of the single Verilog Boolean expression a, or the Verilog Boolean expression {a} (a trivial concatenation). Several suggestions for syntactic changes were made.

   Resolution (for version 1.1): Make no syntactic changes. Rather, define that {a} will always be parsed as a SERE.

2. Tokenisation: Accellera Version 1.01 does not clearly enumerate the tokens of the language.

   Resolution: Appendix A.2 of Accellera Version 1.1 includes a list of tokens.

3. Case-sensitivity: Accellera Version 1.01 states that keywords are case-sensitive in all flavours, but this is inconsistent with the spirit of VHDL for the VHDL flavour.

   Resolution: Accellera Version 1.1 states that keywords are reserved identifiers. Therefore, the case-sensitivity rules of the underlying flavour apply.

4. Operator precedence: The operator precedence table of Accellera Version 1.01 is missing some operators.

   Resolution: Missing operators were added to Accellera Version 1.1.

5. PSL_Identifier: The BNF is missing the definition of PSL_Identifier.

   Status: This is a comment that was submitted in March 2004 and seems to have been lost in the shuffle.

6. Sequence: The syntax definition of sequences in a draft version of LRM 1.1 was inconsistent (different definition in Section 6 and Appendix A).

   Resolution: fixed in Accellera Version 1.1.

7. Sequence operators (&, &&, |): The sequence operators (&, &&, |) have different precedence in Verilog and in PSL.

   Resolution: fixed in Accellera Version 1.1.

8. Suffix implication: There are two forms of suffix implication (the old form r(f) and the new form r |-> f) but Section 6 describes their semantics using different wording. This is confusing.

   Resolution: Accellera Version 1.1 clarifies that the semantics are exactly the same.

9. Length matching: the logical contradiction false is treated differently than the structural contradiction.

   Status: This is a comment that was submitted in March 2004 but resolution was postponed due to the difficulties of reformulating the formal semantics. Since that time some ideas for the direction of a solution have been explored. See Section New recommendations for more details.

10. SERE concatenation: Section 6 of a draft of Accellera Version 1.1 uses "holds" rather than "holds tightly" to describe the semantics.

    Resolution: Fixed in Accellera Version 1.1.

11. Inheritance: Section 5 of a draft of Accellera Version 1.1 is unclear regarding whether or not it is legal to have multiple definitions of the same signal in the transitive closure of all inherited verification units.

    Resolution: Clarified in Accellera Version 1.1.

12. Replicated properties: In a draft of Accellera Version 1.1, a repetition count is required to be statically computable, but on the other hand, it is suggested that a forall Name can be used as a repetition count.

    Resolution: Clarified in Accellera Version 1.1 that the parameter defined by a replicator is considered to be static.

13. Verification directives: In a draft of Accellera Version 1.1, it was not clear what the scope of a label is.

    Resolution: Clarified in Accellera Version 1.1.

14. Core language: It was suggested that perhaps the language is too complex, and that standardization should be split into at least two parts – a small subset to be standardized now, and the rest of the standardization process to be postponed to a later date. Note: this opinion was not held by all members of the PROSYD consortium, but was nevertheless raised for discussion in Accellera.

    Resolution: This proposal was rejected, on the basis that PSL was already a standard at the time it was made, and therefore the proposal was irrelevant.

15. Typed forall: the forall construct introduces a variable whose type is not specified. Consider requiring it to be typed.

    Resolution: This proposal was considered as part of the Group I proposals in the IEEE extensions committee. Ultimately, it was resolved by requiring that each value in the value range be statically computable, and that all be of the same type.

16. Sampling semantics: PSL does not define the sampling semantics, nor does it define how a trace is extracted from the design by a verification tool. It was suggested that PSL explicitly define the sampling semantics, and how a trace is extracted from a verification tool. Note: this opinion was not held by all members of the PROSYD consortium, but was nevertheless raised for discussion in Accellera, as part of the Group M proposals.

    Resolution: Group M was deferred to a future version of PSL.

17. Non-determinism: PSL needs better support for non-determinism, especially in the Verilog and VHDL flavours.

    Resolution: This was examined by the IEEE committee as part of Group J, and a proposal for extended non-determinism support was adopted. The adopted proposal aligns the syntax and semantics of the new built-in functions nondet() and nondet_vector() with those of forall.

New recommendations

In this section we detail new issues that we recommend Accellera and IEEE consider for the next version of the standard. They are:

1. (This is related to item 9 of the previous section.) It has been known since Accellera Version 1.01 (where it is mentioned in Appendix B) that there is an inconsistency in the way that structural contradictions are treated vs. the way that logical contradictions are treated. During the past year some ideas for the direction of a solution have been explored, and are described in [EFH05]. We recommend that a future version of the formal semantics base a solution to this problem on [EFH05].
2. The semantics of r[*0] are different depending on whether or not r[*0] is standalone or is concatenated with another SERE. There is no good reason for this. We recommend that it be changed, and note that the formal semantics of [EFH05] fix this problem as well.
3. The paper “Combining System Level Modeling with Assertion Based Verification”, by A. Dahan, D. Geist, L. Gluhowski, D. Pidan, G. Shapir, Y. Wolfsthal, M.L. Benalycherif, R. Kamdem, Y. Lahbib, 6th International Symposium on Quality Electronic Design, IEEE Computer Society ACM 2005 describes a method of using PSL in conjunction with SystemC. We recommend that a SystemC flavour of PSL be considered.
4. In Accellera Version 1.1, the only supported types for formal parameters of named properties are Boolean, constant, sequence and property. We recommend that the supported types be extended to include Bit, BitVector, Numeric and String.

5. Suppose we would like to say that signal "a" is independent of signal "b". This could be checked by a cone-of-influence reduction. Currently, there is no way to say such meta-properties in PSL. We recommend that inclusion of such language constructs be considered.
6. We recommend that a new directive called "witness" be added. This will instruct a verification tool to produce a trace on which the property holds (as opposed to "assert", which instructs the tool to verify that it holds on all traces).

Future directions

In this section we would like to call the attention of Accellera and the IEEE to the work of Oded Maler and Amir Pnueli, in Deliverable 1.3/1, which discusses the issues involved in extending PSL for analogue circuits. The deliverable gives a preview of what can be expected in the final proposal for language extensions at +36.

3 References

[EFH05] C. Eisner, D. Fisman and J. Havlicek. A topological characterization of weakness. Submitted to PODC 2005.