

IEEE P1076 Ballot Responses

15 July 2008, DRAFT

The role of Ballot Resolution Committee for the P1076 Revision ballot was assigned to the Issues Screening and Analysis Committee (ISAC) of the P1076 Working Group. The ISAC addressed all comments raised by balloters, not just those associated with Disapprove votes. The ISAC is grateful to balloters for their comments, as they help to improve the quality of the revised standard.

The ISAC maintains a repository of issues in a Bugzilla tracking database, accessible at

<https://bugzilla.mentor.com/>

During the ballot process, a number of technical and editorial issues relating to the balloted draft were reported to the ISAC and entered into the database. The ISAC has analyzed the issues and recommended changes. These changes are also detailed in this document and are included in the revised draft, P1076-2008/D4.3. Balloters are invited to access the database to review the ISAC analyses in detail. The annotated version of the revised draft includes footnotes tracing changes back to the entries in the database.

Contents

“Responses to Ballot Comments” on page 2

“Issues from ISAC Tracking Database” on page 15

Responses to Ballot Comments

Ballot comment #1

Name: Ashenden, Peter

Vote: Approve

Category: General

Page:

Clause/subclause:

Line:

Must be satisfied: No

Comment

A number of minor issues have been identified and reported in the ISAC issue tracking database. The changes recommended by ISAC should be incorporated.

Proposed Resolution

Incorporate the changes proposed by ISAC.

Resolution

Status: Accepted

The changes proposed by ISAC are detailed later in this document.

Ballot comment #2

Name: Coordination, Scc14

Vote:

Category: General

Page:

Clause/subclause:

Line:

Must be satisfied: Yes

Comment

SCC14 coordination: OK

Proposed Resolution***Resolution***

Status: Accepted

No action required.

Ballot comment #3

Name: Steinhagen, Jennie M

Vote:

Category: Editorial

Page: 7

Clause/subclause: 2

Line:

Must be satisfied: Yes

Comment

Normative references are those normative documents that contain material that must be understood and used to implement the standard. Thus, referenced documents are indispensable when applying the standard. Each normative reference shall be cited, and the role and relationship of each referenced document shall be explained in the body of the standard.

Proposed Resolution

It appears that IEEE Std 754 is in both the norm refs and the biblio. It should only be in one or the other.

Resolution

Status: Accepted

See also "Bugzilla issue #245" on page 34

Further examination shows that references to 754, 854, UML and several "C" references occur in both places. In accordance with IEEE standards, these references should be removed from the bibliography and corresponding "reference labels" in the text should be redirected to the normative section.

Deleted the following from Annex J: [B2], [B3], [B10], [B11], [B12], [B13], [B14], [B33].

In Clause 2, Normative references, replaced reference to OMG UML specification with:

ISO/IEC 19501:2005, Information technology—Open Distributed Processing—Unified Modeling Language (UML) Version 1.4.2.

since this is the latest ISO version, and the VHDL spec doesn't use any features from later OMG versions.

Adjusted references in text to deleted bibliography entries by removing cross ref and citing designation of the referenced standard. In the first mention of a normative reference (in 1.6.5), added a footnote with the text "For information on references, see Clause 2", as specified in the IEEE Standards Style Manual.

Ballot comment #4

Name: Hanna, William

Vote: Disapprove

Category: General

Page:

Clause/subclause:

Line:

Must be satisfied: Yes

Comment

All references to VHPI should be made optional - should be removed from body of the LRM into an Appendix.

Proposed Resolution

Move VHPI to an Appendix - Annex and state that it is optional

Resolution

Status: Rejected

See also "Bugzilla issue #244" on page 34

The VHPI was incorporated into VHDL as an amendment, 1076c. Its inclusion as an integral, non-optional part of the standard was approved by the balloting body for the amendment project, and the amended standard was approved by IEEE in 2007.

Specification of the VHPI requires some significant changes to details of aspects of the language, notably, to the simulation cycle as specified in Clause 12 of IEEE 1076c-2007. Making the VHPI optional would require specifying and maintaining two versions of the simulation cycle, one without VHPI and one with VHPI. This would be difficult and error prone.

The ISAC notes that a minimal implementation of the VHPI need not be burdensome. The VHPI provides a means for a tool to provide a subset of the full VHPI capability and for VHPI programs to query the capabilities provided by the tool (see subclause 17.3 of D4.2). A minimal implemen-

tation need only provide the function interface described in Clause 23 and Annex B, with none of the capabilities described in 17.3 being implemented by the tool. Calls to the functions would generally return an error indication.

To make this explicit, the following note is added to subclause 17.3:

NOTE—A minimal implementation of the VHPI need only provide the function interface described in Clause 23 and Annex B, with none of the capability sets described in this subclause being implemented by the tool. In such a minimal implementation, calls to functions would, in most cases, raise a VHPI error condition.

Ballot comment #5

Name: Karocki, Piotr

Vote: Approve

Category: Editorial

Page: 195

Clause/subclause: 10.2

Line: 47

Must be satisfied: No

Comment

that wait until True;

Proposed Resolution

I'm not sure, but I think it should be
that wait until TRUE;

Resolution

Status: Accepted

See also “Bugzilla issue #248” on page 36

The case of the identifier is changed as proposed. In reviewing this comment, the ISAC found other occurrences of the identifiers FALSE and BOOLEAN that were in title case rather than upper case. They have also been changed. The ISAC notes that this is an editorial correction, as the case of identifiers is not significant in VHDL.

Ballot comment #6

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 63

Clause/subclause: 5.7

Line: 54

Must be satisfied: No

Comment

"The sequence of characters of abstract literal.." should read "The sequence of characters of the abstract literal.."

Proposed Resolution

Fix text as proposed

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed as proposed.

Ballot comment #7

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 77

Clause/subclause: 6.5.2

Line: 58

Must be satisfied: No

Comment

"..monitors and checkers, it not feasible.." should read "..monitors and checkers, it is not feasible.."

Proposed Resolution

Fix text as proposed

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed as proposed.

Ballot comment #8

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 88

Clause/subclause: 6.5.7.2

Line: 20

Must be satisfied: No

Comment

On this line and the next 7, spacing is applied inconsistently

Proposed Resolution

Apply spacing consistently

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed as proposed.

Ballot comment #9

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 115

Clause/subclause: 8.6

Line: 2

Must be satisfied: No

Comment

"For such an attribute that denotes a function," should read ""For an attribute that denotes a function,"

Proposed Resolution

Fix text as proposed

Resolution

Status: Accepted

See also “Bugzilla issue #246” on page 34

The intention was to refer to an attribute that is a predefined attribute. However, the resulting sentence construction is clumsy. As only predefined attributes can denote functions, it would be best to delete the word “such” to make the sentence clearer. Fixed in this way.

Ballot comment #10

Name: Christen, Ernst

Vote: Approve

Category: Technical

Page: 122

Clause/subclause: 9.2.2

Line: 17

Must be satisfied: No

Comment

The text on lines 17-51 uses both "bits" and "elements" to refer to the elements of R, but "bits" is never defined..

Proposed Resolution

Replace "bits" by "elements". Alternatively, provide a definition for "bit".

Resolution

Status: Accepted

See also “Bugzilla issue #247” on page 35

All six occurrences of “bits” replaced with “elements”.

Ballot comment #11

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 122

Clause/subclause: 9.2.2

Line: 55

Must be satisfied: No

Comment

"The unary logical operators belongs.." should read The unary logical operators belong.."

Proposed Resolution

Fix text as proposed

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed as proposed.

Ballot comment #12

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 237

Clause/subclause: 15.8

Line: 64

Must be satisfied: No

Comment

"..by concatenating the to left of the expanded bit.." should read "..by concatenating to left of the expanded bit.."

Proposed Resolution

Fix text as proposed

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed as proposed.

Ballot comment #13

Name: Christen, Ernst

Vote: Approve

Category: Technical

Page: 239

Clause/subclause: 15.10

Line: 49

Must be satisfied: No

Comment

"The following identifiers are called reserved words and are reserved for significance in the language." but some of these identifiers are reserved because they have significance in PSL, not in VHDL. No definition for the use of these reserved words can be found in the text of the document, this is in [B33].

Proposed Resolution

Identify PSL reserved words as such and add an explanation stating that the use of these reserved words is defined in [B33]

Resolution

Status: Accepted

See also "Bugzilla issue #243" on page 33

Added the following note to 15.10

NOTE 3—The following reserved words are PSL keywords, that is, reserved identifiers in PSL:

assert	default	restrict_guarantee	vprop
assume	fairness	sequence	vunit
assume_guarantee	property	strong	
cover	restrict	vmode	

Their use in PSL is defined in [B33]. Other PSL keywords, reserved only within PSL declarations, PSL directives and PSL verification units, are defined in [B33].

Ballot comment #14

Name: Christen, Ernst

Vote: Approve

Category: Editorial

Page: 241

Clause/subclause: 15.10

Line: 9

Must be satisfied: No

Comment

"The reserved word range is also used as the name of a predefined attribute." So is "subtype".

Proposed Resolution

Fix note to include "subtype" as a reserved word that is also a predefined attribute

Resolution

Status: Accepted

See also "Bugzilla issue #246" on page 34

Changed to "The reserved words range and subtype are also used ..."

Ballot comment #15

Name: Navabi, Z

Vote: Approve

Category: Technical

Page: 184

Clause/subclause:

Line: 40

Must be satisfied: No

Comment

Extension of certain constructs like the generate statement to include case makes the language unnecessarily complex without much use for describing actual hardware.

Proposed Resolution

Resolution

Status: Rejected

See also "Bugzilla issue #250" on page 37

The extensions to the generate statement, along with other extensions, were based on requirements gathered from users. The Accellera VHDL Technical Committee analysed and prioritized the requirements, and developed language extensions to meet them. A review subcommittee validated the extensions against the requirements, and then an LRM editing subcommittee developed the detailed LRM edits to implement the extensions. Throughout this process, members of the VHDL community were repeatedly invited to participate and provide further input. At no stage did any participant argue against extending the generate statement. Hence, the committee believes the extension remains a priority for users.

Ballot comment #16

Name: Schwarm, Stephen

Vote: Approve

Category: Editorial

Page: iv

Clause/subclause: Introduction

Line:

Must be satisfied: No

Comment

"new introduction" needs some content. I suspect that is was a place holder

Proposed Resolution

I think much of what was in the original text was good and jut needs editing.

Resolution

Status: Accepted

See also "Bugzilla issue #251" on page 37

Text of new introduction:

The VHSIC Hardware Description Language (VHDL) 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.

This document specifies IEEE Std 1076™-2008, which is a revision of IEEE Std 1076™c-2007. Initial work on gathering requirements and developing language extensions was undertaken by the IEEE VHDL Analysis and Standardization Group (VASG), otherwise known as the 1076 Working Group. Subsequently, Accellera (www.accellera.org) sponsored an effort to complete that work and draft a revised Language Reference Manual. That draft was returned to IEEE for final revision and approval, resulting in this document and the associated machine-

readable files. This revision incorporates numerous enhancements, both major and minor, to previously existing language features and several new language features. The changes are summarized in Annex E. In addition, several VHDL library packages that were previously defined in separate standards are now defined in this standard, ensuring that they are treated as integral parts of the language. Finally, this revision incorporates the IEEE Property Specification Language (PSL) as part of VHDL. The combination of these changes significantly improves VHDL as a language for specification, design, and verification of complex electronic systems.

The maintenance of the VHDL language standard is an ongoing process. The chair of the VHDL Analysis and Standardization Group extends his gratitude to all who have participated in this revision, both in the IEEE committees and the Accellera effort, and encourages the participation of all interested parties in future language revisions. If interested in participating, please contact the VASG at stds-vasg@ieee.org or visit the following website: <http://www.eda.org/vasg>.

Ballot comment #17

Name: Dovich, Steven

Vote: Approve

Category: Editorial

Page: 525

Clause/subclause: 24.1.1

Line: 10

Must be satisfied: No

Comment

The phrase "This allow an author of a VHDL description" reads badly.

Proposed Resolution

Change to "This allows and author of a VHDL description"

Resolution

Status: Accepted

See also "Bugzilla issue #252" on page 37

Changed to "This allows an author..."

Ballot comment #18

Name: Dovich, Steven

Vote: Approve

Category: Technical

Page: 525

Clause/subclause: 24.1.1

Line: 18

Must be satisfied: No

Comment

Inconsistency in terminology use leaves the text confusing. Specifically the use of the term decryption where description would be correct.

Proposed Resolution

Change the sentence:

"An encryption envelope is a portion of the description that is to be encrypted, and a decryption envelope is a portion of the decryption containing previously encrypted text to be decrypted."

to read:

"An encryption envelope is a portion of the description that is to be encrypted, and a decryption envelope is a portion of the description containing previously encrypted text to be decrypted."

Resolution

Status: Accepted

See also "Bugzilla issue #253" on page 38

This typo was previously noted, and the text has been revised. See "Bugzilla issue #217" on page 20.

Issues from ISAC Tracking Database

Bugzilla issue #62

Summary: Matching operators not needed for BOOLEAN

Category: Technical

Description

The removal of BOOLEAN versions of matching operators was incomplete: they are still listed on page 327 of D4 (listing of package STANDARD).

Resolution

In an earlier draft of the revised standard, the matching relational operators were predefined for type BOOLEAN. Further analysis of requirements determined that they should not be predefined, and so their definition was removed from 9.2.3. However, their mention on page 259 of D4.2 was not removed. This is corrected in D4.3.

Bugzilla issue #208

Summary: D4.2 simple typos

Category: Editorial

Description

Numerous typographical errors and minor editorial corrections.

Resolution

Changed as listed in the database at https://bugzilla.mentor.com/show_bug.cgi?id=208.

Bugzilla issue #210

Summary: ?? for STD_ULONGIC still listed in table in 9.2.9

Category: Technical

Description

In 9.2.9, description of the ?? operator for std_ulongic was supposed to have been removed. However, there is still a row for it in the table.

Resolution

The row has been deleted.

Bugzilla issue #211

Summary: Confusing rules for object aliases.

Category: Editorial

Description

This issue is also present in VHDL-2002.

The clean version of D4.2 page 92 line 29 says:

"d) The same applies to attribute references where the prefix of the attribute name denotes the alias."

Its not clear (to me) what is meant by "the same". This rule could be better stated.

Resolution

Changed d) to

d) When the prefix of an attribute name denotes the alias defined by the alias declaration, sub-rules 1), 2), and 3), of rule c) apply.

Bugzilla issue #212

Summary: Inconsistent treatment for PSL declarations

Category: Technical

Description

Clause 6.1 gives a list of declarations.

This list includes PSL property declaration and PSL sequence declaration but does not include PSL clock declaration.

Conversely, the rest of clause 6 gives no mention to PSL property declarations or PSL sequence declarations, but has a subclause (6.11) on PSL clock declarations. Its not necessary to add sub-clauses on PSL property declarations or PSL sequence declarations, but we should add PSL clock declaration to the list in 6.1.

Resolution

Added the following note to 6.11:

NOTE—A PSL clock declaration differs from other declarations in VHDL and PSL in that it does not declare a designator denoting some entity. It is more akin to a VHDL specification in that it associates additional information with PSL directives within a design. Hence, it is not listed as a declaration in 6.1. Since it is called a declaration in the PSL standard [B36], it is included in this clause for ease of reference, rather than in Clause 7.

Bugzilla issue #213

Summary: Can a force signal assignment update ports of mode IN?

Category: Technical

Description

D4.2 clean version of the LRM states on page 77 that a signal is updated when it is the target of an assignment. On the same page it states that interface objects of mode IN shall not be updated. In addition page 383 states:

"It is an error if a VHPI program updates the value of a formal parameter of mode IN." This last sentence might not be applicable to the issue because it might be interpreted as dealing only with the `vhpi_put_value` function.

The problem is that on page 153 it states, in giving the rules for force semantics (which is a form of assignment):

"If the target is a port or signal parameter of mode IN, a force mode of IN is used."

I conclude that the target can never (legally) be of mode IN, so this rule makes no sense.

Resolution

The definition of updating an interface object (page 77 of clean D4.2) is extended by inserting the following list items after the first list item:

- When a VHPI information model object representing the given object is updated using a call to the function `vhpi_put_value`
- When the object is a signal and the `vhpi_schedule_transaction` function is used to schedule a transaction on a driver of the signal

The rule of an in-mode interface object (page 77 of clean D4.2) is extended to the following:

- in. The value of the interface object is allowed to be read, but it must shall not be updated by a simple waveform assignment, a conditional waveform assignment, a selected waveform assignment, a concurrent signal assignment, or a variable assignment.

(The various forms of assignment involving waveforms are listed, since we don't define a "waveform assignment" anywhere.)

Added the following note to the end of the subclause:

NOTE 8--A port of mode in may be updated by a force assignment, a release assignment, or a call to `vhpi_put_value`. A formal parameter of mode in shall not be updated by a call to `vhpi_put_value` (see 22.5.1).

Bugzilla issue #214

Summary: Semantics of some READ procedures are not complete

Category: Technical

Description

This problem already existed in VHDL-2002 but was discovered while examining the semantics of the new SREAD procedure. This problem exists for forms of the READ procedure as well as for the new procedures OREAD and HREAD and SREAD. Certain READ procedures, such as READ[BIT_VECTOR] require a specified amount of data--it is an error if VALUE'LENGTH characters are not read and reading stops if that number of characters are read. However, most READS, including SREAD, HREAD, and OREAD read a variable number of characters until "a character is encountered that cannot be part of the value according to the rules for ..." What's missing is the very common case in which all the data on the line is read. So, if interpreting the rules literally a READ[INTEGER] on the string "1234 " is legal, but the same READ in "1234" is not defined, because there is no stopping rule provided.

In practice, every tool behaves as expected, but the rules should be correct.

Resolution

Changed paragraphs as follows:

Page 276 of clean D4.2, line 54:

For all READ procedures, character removal and string composition stops when the end of the line is encountered. Character removal and string composition also stops when a character is encountered that cannot be part of the value according to the rules for string representations, or, in the case of the READ procedure for BIT_VECTOR, is not an underline character that can be removed according to the preceding rule; this character is not removed from L and is not added to the string representation of the value.

Page 277 of clean D4.2, line 8:

Character removal and string composition stops when the end of the line is encountered. Character removal and string composition also stops when a whitespace character is encountered or VALUE'LENGTH characters have been accepted; the whitespace character is not removed from L and is not added to the string.

Page 277 of clean D4.2, line 20:

Character removal and composition stops when the end of the line is encountered. Character removal and string composition also stops when a character is encountered that is not an octal (respectively, hexadecimal) digit or an underline character that can be removed according to the preceding rule; this character is not removed from L and is not added to the string. Moreover, character removal and composition stops when the expected number of digits have been removed, ...

Bugzilla issue #215

Summary: Problem with deallocation of lines in textio.

Category: Technical

Description

This problem exists in VHDL-2002 as well as D4.2. Clean version of D4.2 on page 278 in Note 2 states that a read or write may modify "or even deallocate" the string object referenced by the line. However, I was unable to find normative text which implied this.

The LRM does state that under certain conditions a READLINE deallocates a non-null line, but that is the only time deallocation is mentioned in this clause. The LRM does say that READ commands modify the string value designated by L and that WRITELINE returns a null string value.

I believe that the comment is correct and that no other treatment would be satisfactory, the the LRM doesn't actually contain normative text to that effect.

Resolution

LRM changes (wrt D4.2 clean) are:

Page 276, lines 11-13: Change the text describing READLINE

If parameter L contains a non-null access value at the start of the call, the object designated by that value is deallocated before the new object is created.

to

If parameter L contains a non-null access value at the start of the call, the procedure may deallocate the object designated by that value.

Page 276, line 20: Insert the following sentence describing WRITELINE and TEE before the last sentence in the paragraph:

If parameter L contains a non-null access value at the start of the call, the procedures may deallocate the object designated by that value.

Page 276, line 48: Add the following to the paragraph describing the READ, SREAD, OREAD, and HREAD procedures:

Each procedure may modify the value of the object designated by the parameter L at the start of the call or may deallocate the object.

Page 277, lines 36-37: delete the following from the text describing WRITE:

in this case, however, L continues to designate the entire line after the value is appended

Page 277, line 45: insert a new paragraph describing WRITE, OWRITE, and HWRITE

For each WRITE, OWRITE and HWRITE procedure, after data is appended to the string value designated by the parameter L, L designates the entire line. The procedure may modify the value of the object designated by the parameter L at the start of the call or may deallocate the object.

Page 276, lines 40-42: Revised Note 2 to the following:

NOTE 2---Since the execution of a read or write operation may modify or deallocate the string object designated by input parameter L of type Line for that operation, a dangling reference may result if the value of a variable L of type Line is assigned to another access variable and then a read or write operation is performed on L.

Bugzilla issue #216

Summary: Clause title on VHDL-87 is out of date.

Category: Editorial

Description

Clause 16.8.5.3 is titled: Compatibility with IEEE std 1076-1987

However, the clause actually deals with compatibility of the proposed standard D4.2 with all previous versions. I suggest that the title of this clause be changed to something more generic such as:

"Compatibility with previous versions of IEEE Std 1076".

Resolution

Changed to "Compatibility with previous editions of IEEE Std 1076"

Bugzilla issue #217

Summary: Confusing/contradictory requirements for viewport objects

Category: Technical

Description

D4.2 clean version says on page 441(453 lines 37-38:

"The declared object is declared in the source text of the protection envelope containing the protect viewport directive."

However, on the same page lines 49-51 states:

"The string literal following the access keyword specifies which operations are permitted for the identified declared object or objects, regardless of whether they are declared in a portion of the VHDL description that is included in a protection envelope, as follows:"

I suggest that we remove the phrase "regardless of ...protection envelope"

Resolution

In 24.1.1, changed the text to

An encryption envelope contains protect directives and a portion of the description, called the source text, that is to be encrypted. A decryption envelope contains protect directives and previously encrypted text to be decrypted.

In 24.1.2.23, deleted the phrase "regardless of ...protection envelope".

Bugzilla issue #218

Summary: Permitted access into encrypted code unclear

Category: Technical

Description

What access is allowed to objects in encrypted text?

D4.2 clean version states in clause 24.1.2.23 on viewports that if a viewport object has read-only access then

"The object may be read by any part of the VHDL description, by a VHPI program, or by a tool. It is an error if the object is updated other than by part of the VHDL description contained within a protection envelope." Similar wording applies to write-only access. These paragraphs imply that you can read and write encrypted objects from encrypted text.

Furthermore, on pages 450(462 and 451(463 it says:

"If a decryption tool provides a means for the tool user to gain access to a representation of a VHDL description or an elaboration of a VHDL description (for example, by means of a user interface or an applications programming interface such as VHPI), then the tool shall not provide access to any representation of a decrypted portion of a VHDL description other than a representation of an object specified by a protect viewport directive."

So it is clear that GUI access and VHPI access into encrypted regions are not allowed. But what about access from VHDL code into encrypted regions?

The section on viewports implies that access from one encrypted region into another is allowed and suggests that access from a non-encrypted region into an encrypted region is not allowed.

Full solution to this issue will probably require significant changes to the overall specifications for protected regions, but in the short term it would be good to clarify the intended allowable access.

Resolution

In 24.1.2.23, changed the list items describing read-only and write-only access to:

-- "R": Read-only access— ... It is an error if the object is updated other than by part of the VHDL description contained within the protection envelope containing the protect viewport directive.

-- "W": Write-only access— ... It is an error if the object is read other than by part of the VHDL description contained within the protection envelope containing the protect viewport directive.

Bugzilla issue #220

Summary: Inconsistent use of vhpIntT and int

Category: Technical

Description

(clean version) page 402 states:

```
int vhpi_get(vhpiIntPropertyT property, vhpiHandleT object);
```

and

```
vhpiIntT vhpi_get_cb_info (vhpiHandleT object, vhpiCbDataT *cb_data_p);
```

However, page 475 has

```
XXTERN vhpiIntT vhpi_get (vhpiIntPropertyT property, vhpiHandleT object);
```

and

```
XXTERN int vhpi_get_cb_info (vhpiHandleT object, vhpiCbDataT *cb_data_p);
```

For consistency with other functions, these should both probably return int.

Resolution

Upon closer inspection, the way `vhpi_get` is declared in `vhpi_user.h` is consistent with the other property access functions, `vhpi_get_str`, `vhpi_get_real`, and `vhpi_get_phys`. These each return the value of the property as their result. Compare this with other functions, which return information through memory pointed to by a parameter and return a status value as the int result. So it looks like the definition of `vhpi_get` as having `vhpiIntT` for its return type is intended. I've updated the declaration of `vhpi_get` and `vhpi_get_cb_info` in the LRM to be consistent with the header file.

Bugzilla issue #221

Summary: Unclear descriptions of `VHPI_SENS_ISSET` and `VHPI_SENS_FIRST`

Category: Technical

Description

Clauses B.2.5 `VHPI_SENS_ISSET` and B.2.6 `VHPI_SENS_FIRST` have the same descriptions:

Determines whether a bit in a sensitivity-set bitmap is set.

I suggest for the first

Determines whether a specific bit in a sensitivity-set bitmap is set.

and for the second:

Determines whether any bit in a sensitivity-set bitmap is set.

Resolution

Fixed.

Bugzilla issue #222

Summary: condition expression should be boolean

Category: Technical

Description

I think that this issue was raised before--I ask that it be reconsidered. Pre 200X syntax for the condition grammar rule was:

```
condition ::= boolean_expression
```

In conjunction in implicit application of the condition operator this was changed to:

```
condition ::= expression
```

I believe that the original expression is superior and should be kept for this reason: consider the following analogous situation:

```
variable v1: integer;  
...  
v1 := 1;
```

The type of v1 is (anonymous base type of) INTEGER. The type of "1" is universal_integer. The rules for variable assignment require that the type of the expression be the same as the type of the target. Since the types don't match, there is a rule for implicit conversion.

Similarly, the type of the condition expression must be BOOLEAN. Since the types don't match there is rule for an implicit conversion (via the ?? operator).

In addition the places in which the condition operator is applied could be simplified. Most of them would be covered by something like:

"as the boolean expression in a condition "

Resolution

The proposal, as submitted, was rejected at a meeting held on 05 June 2008. However, it was decided to change "guard expression" with "guard condition" to reflect the boolean implication of "condition".

Guard expression changed to guard condition in numerous places, including index markers.

Bugzilla issue #223

Summary: Contradictory statements about normative status of 1164 body.

Category: Technical

Description

Clean version page 453 states:

"The bodies of the standard multivalued package and the standard synthesis packages, on the other hand, are normative specifications of the semantics of the packages."

However on page 520 it states:

"The package body is intended to provide a guideline for implementors. It is not a normative part of this standard, but suggests a way in which implementors may implement the STD_LOGIC_1164 package."

I believe that the first statement is correct, since there are some details/ corner cases which can only be deduced by reading the package implementations.

Resolution

Deleted the paragraph on page 520 (clean version), lines 31 to 36. The paragraph on page 280, lines 44 to 50, normatively specify that the package body is normative.

Bugzilla issue #224

Summary: In glossary, some references are out of order.

Category: Editorial

Description

Because of movements of text passages, glossary entries have multiple references which are now out of order. For example, I.10 actual, 6.5.7.1 precedes 6.5.6.2. It would be nice, although not essential, if the references were in order.

Resolution

Fixed the occurrence mentioned and others.

Bugzilla issue #225

Summary: Glossary description of complete context is no longer correct

Category: Technical

Description

Glossary entry I.61 complete context: fails to mention that certain ranges and expressions are also complete contexts, (as a result of recent LRM changes).

Resolution

Changed to "A declaration, a specification, or a statement, or, in certain cases, a discrete range or an expression; ...". The cross-reference to 12.5 identifies the cases.

Bugzilla issue #226

Summary: glossary entry for "design unit" is no longer correct

Category: Technical

Description

Now that there are more primaries, glossary entry I.94 "design unit:" needs to be updated.

Resolution

Fixed: "A design unit is either an entity declaration, an architecture body, a configuration declaration, a package declaration, a package body, a package instantiation declaration, a context declaration, or a PSL verification unit."

Bugzilla issue #227

Summary: glossary entry for expanded name has incorrect reference

Category: Technical

Description

Glossary entry I.131 "expanded name" indicates a reference to 10.2. 10.2 is "Wait Statements" and there is no such reference. (In VHDL-93 there was such a reference, but it was removed in VHDL-2002, and if the phrase on "selected names" is interpreted as syntactic selected names, it doesn't need to be added back.)

Resolution

At a committee meeting held on 19 June 2008 It was noted that expanded names were removed from draft D2 of the 2002 version, in response to a comment from Paul Graham who pointed out that the prefix of an expanded name is not a signal. However, the decision to drop the phrase was not correct, since a signal can be indicated by means of an expanded name. Instead the phrase in the Wait Statement Clause of VHDL-93 "An expanded name whose prefix denotes a signal..." should be changed to "An expanded name that denotes a signal..." and added back into D4.2.

Reinstated and revised the list item as agreed.

Bugzilla issue #228

Summary: Glossary entry for floating-point type is incorrect

Category: Technical

Description

Glossary entry I.141 "floating-point types" states:

"A discrete scalar type..."

floating-point types are not discrete.

Resolution

Also incorrect is the statement that the representation includes at least six digits of precision. The current requirement is IEEE 754 or 854 format with at least 64 bits. I've dealt with both errors by changing the text to:

A scalar type whose values approximate real numbers. The representation of a floating-point type conforms either to IEEE Std 754-1985 [B2] or to IEEE Std 854-1987 [B3] and has a minimum size of 64 bits.

Bugzilla issue #229

Summary: Is type `vhpiIntT` signed or unsigned?

Category: Technical

Description

clause 22.2.3 states that the type `vhpiIntT` is at least 32 bits and

"The value represented by a given value of either type is the given value, interpreted as a signed twos-complement binary number."

However, the header file has:

```
typedef uint32_t vhpiIntT;
```

Which is correct?

A similar problem exists for type `vhpiLongIntT`

Resolution

In the ISAC telecon on 19-June, it was agreed that `vhpiIntT` and `vhpiLongIntT` should be signed. This is consistent with the description in 22.2.3.

It was also noted that `vhpiCharT` should be unsigned to be consistent with 22.2.4. As it is in the header file, it is just `char`, which may be signed or unsigned depending on the implementation. Changed to unsigned `char`.

Further, `vhpiSmallPhysT` is changed to signed, to be consistent with 22.2.6.

The revised declarations are:

```
typedef int32_t vhpiIntT;
typedef int64_t vhpiLongIntT;
typedef unsigned char vhpiCharT;
...
typedef int32_t vhpiSmallPhysT;
```

Bugzilla issue #231

Summary: process(all) -- compiler cannot infer the sensitivity list

Category: Technical

Description**11.3 Process statement**

It is an error if a process statement with the reserved word `all` as its process sensitivity list is the parent of a subprogram declared in a design unit other than that containing the process statement and the subprogram reads an explicitly declared signal that is not a formal signal parameter or member of a formal signal parameter of the subprogram or of any of its parents. Similarly, it is an error if such a subprogram reads an implicit signal whose explicit ancestor is not a formal signal parameter or member of a formal parameter of the subprogram or of any of its parents.

The compiler can only check the first part of the condition, i.e.:

It is an error if a process statement with the reserved word `all` as its process sensitivity list is the parent of a subprogram declared in a design unit other than that containing the process statement.

The compiler does not know bodies of referenced subprograms, so it cannot check which signals are read by the subprogram. Here is an example:

[...]

While doing an elaboration time check for the example I provided is technically feasible that would, IMHO, violate the language library system rules -- recompiling a package body affects the design unit that references the package. (Note that the package body is `_not_` referenced in the design unit -- it is the package that is referenced.)

[...]

Resolution

The rules for process(all) introduce package body dependencies only if the error conditions described are required to be detected during analysis of the design unit. It has always been recognized that error checking for process(all) semantics cannot be completed by the analyzer. (Similar issues exist for checking hierarchical references.) To emphasize this important observation, a note will be added to the LRM stating that the semantic checks for process (all) cannot, in general, be done at analysis time.

Added the following to 11.3:

NOTE 4---In general, it is not possible to determine at analysis time whether a process with the reserved word `all` as its process sensitivity list is the parent of a subprogram declared in a separate design unit and whether the rules for such a subprogram are met.

Bugzilla issue #233

Summary: Vhpi_user.h requires enum types contents update/completion

Category: Technical

Description

vhpiClassKindT enum type does not contain values for concrete classes: concProcCallStmt and seqProcCallStmt instead of abstract class procCallStmt. Similar problem for concAssertStmt and the like.

Proposed Resolution

"vhpi_user.h" requires update and synchronization with VHPI information model. For the classes concProcCallStmt and seqProcCallStmt this would be adding vhpiConcProcCallStmtK and vhpiSeqProcCallStmtK to vhpiClassKindT enum type.

Resolution

In vhpiClassKindT, added enum values for the following concrete classes:

- concAssertStmt
- concProcCallStmt
- foreverLoop
- seqAssertStmt
- seqProcCallStmt
- seqSigAssignStmt

In vhpiClassKindT, marked the following enum values for abstract classes as deprecated:

- vhpiAssertStmtK
- vhpiLoopStmtK
- vhpiProcCallStmtK

In vhpiClassKindT, the enum value vhpiProtectedTypeK should be vhpiProtectedTypeInstK. This is done by deprecating vhpiProtectedTypeK and adding vhpiProtectedTypeInstK.

In vhpiClassKindT, the enum value vhpiOperatorK does not correspond to any class and is marked deprecated.

In vhpiOneToOneT, the following do not correspond to any associations and are marked deprecated:

- vhpiBasicSignal
- vhpiCurEqProcess
- vhpiDecl
- vhpiIterScheme
- vhpiName
- vhpiOperator
- vhpiParamExpr

In vhpiOneToOneT, enum values for the following 1-1 associations are added:

Signal
 LibraryDecl
 SimNet
 AliasedName
 CompDecl
 ProtectedTypeInst
 GenIndex

In `vhpiOneToManyT`, the following do not correspond to any associations and are marked deprecated:

`vhpiCondExprs`
`vhpiContributors`
`vhpiEntityClassEntrys`
`vhpiSensitivitys`
`vhpiSpecNames`

In `vhpiOneToManyT`, enum values for the following 1-many associations are added:

`GenerateStmts`
`LocalContributors`
`OptimizedContributors`
`ParamExprs`
`EqProcessStmts`
`EntityClassEntries`
`Sensitivity`

In the information model, the association `Sentivity` is renamed to `Sensivities` so that it is plural, consistent with naming for other 1-many associations.

Added the following for backward compatibility:

```
/* Note: The following macro is defined for compatibility with
   * prototype implementations that use the incorrectly spelled
   * enumeration value. The macro is deprecated and will be removed
   * in a future revision of the standard.
   */
#define vhpiSensitivitys DEPRECATED_vhpiSensitivitys
```

For integer/boolean properties, the `vhpiForeignKindP` enum value does not correspond to any property. However, there is a `Foreign` property for which there is no enum value. This is addressed in Bugzilla issue #235.

Also, for integer/boolean properties, enum values are added for the following:

`AutomaticRestore`
`CompInstKind`
`IsBuiltIn`
`IsDynamic`
`IsOperator`
`NumFields`

For string properties, `vhpiOpNameP` has no corresponding property and is marked deprecated. Enum values are added for the following:

- `CompInstName`
- `InstNames`
- `SignatureName`
- `SpecName`

For physical properties, an enum value is added for the following:

- `Time`

Bugzilla issue #234

Summary: Backward incompatibility for `std_logic_textio` selected names

Category: Technical

Description

At the ISAC telecon on 19 June, it was pointed out that the way the `std_logic_textio` package is defined introduces an incompatibility with the prior version defined by the donor. The procedures defined in the prior version are now defined in `std_logic_1164`, and `std_logic_textio` is empty. The premise is that users will have a use class that uses ll of both packages, viz

```
use ieee.std_logic_1164.all, ieee.std_logic_textio.all;
```

Under this scenario, the placement of the declarations does not matter, as they will become directly visible regardless.

However, if a users writes selected names for the procedures that were in `std_logic_textio`, the difference in placement is exposed. The models will no longer work.

A remedy was suggested, taking advantage of the revised semantics of aliases and homographs. The procedure declarations in the prior version of `std_logic_textio` can be replaced by aliases designating the corresponding declarations in the new version of `std_logic_1164`. That way, selected names for `std_logic_textio` procedures will still work. In the more usual scenario of a use clause referring to all of both packages, each alias and its designated declaration both become directly visible, but there is no ambiguity as they both denote the same named entity.

Resolution

Revised `std_logic_textio` package using aliases

This revision of the package defines aliases for `READ`, `WRITE`, `OREAD`, `OWRITE`, `HREAD`, and `HWRITE`. Aliases are only defined for profiles including `std_ulogic_vector`, not for `std_logic_vector`, since in VHDL-2008, `std_logic_vector` is a subtype of `std_ulogic_vector`.

Bugzilla issue #235

Summary: Error in name of property of foreignf class

Category: Technical

Description

In 19.5, the `foreignf` class has a property named `ForeognT` of type `ForeignT`. There are two problems. First, there is the typo. Presumably `ForeognT` should be `ForeignT`. Second, with that correction, the property name is then the same as the type name. Apart from being confusing, it is inconsistent with naming of properties in other classes. The convention for a property that is of an enum type is to name the property `Blah` and the enum `BlahT`. Examples are `Staticness` of type `StaticnessT` in class `expr`, `State` of type `StateT` in class `callback`, `SigKind` of type `SigKindT` in class `sigDecl`, and others.

The property of class `foreignf` should be renamed to `Foreign`.

Resolution

As it turns out, there is an enum value `vhpiForeignKindP` that doesn't correspond to any property, and the enum value for the `Foreign` property is missing. The name `ForeignKind` for the property makes better sense. The documentation for it is:

The value of type `ForeignT` corresponding to the kind of foreign model or application.

Changed the property name to `ForeignKind`, the enum type to `vhpiForeignKindT`, and kept the `vhpiForeignKindP` enum value.

Bugzilla issue #236

Summary: Wrong case for `specName` property of `blockConfig` class

Category: Editorial

Description

In 19.5, page 315 of D4.2 clean, Figure 3, the `blockConfig` class is shown as having the `specName` property. The property name should start with a capital letter (`SpecName`) for consistency with other property names.

Resolution

Change implemented

Bugzilla issue #237

Summary: Representation of negative time values with `vhpiTimeT`

Category: Technical

Description

22.2.7 specifies that type `vhpiTimeT` represents a non-negative time. 22.2.8 describes value structures and formatting, and specifies that the format `vhpiTimeVal` represents a time value using a time structure of type `vhpiTimeT`.

22.4 further described formatting values and specifies that the native format for objects of VHDL type `TIME` is `vhpiTimeVal`. 22.4 specifies that a tool shall support reading a value using its native format.

The problem is that VHDL objects of type `TIME` may be negative, but the VHPI representation does not allow for negative values. This is an inconsistency.

Note that, since `TIME` is a physical type, an implementation may support reading a `TIME` object using the `vhpiPhysVal` format, which does allow for negative values. However, an implementation is not required to support that format for reading `TIME` values.

A solution to this problem would be to change the definition of `vhpiTimeT` to allow representation of negative time values.

Resolution

In `vhpi_user.h`, changed definition of `vhpiTimeT` to

```
typedef struct vhpiTimeS
{
    int32_t high;
    uint32_t low;
} vhpiTimeT;
```

In 22.2.7, changed the first two paragraphs to:

A value of type `vhpiTimeT` is called time structure and represents a time value. The position number of a time structure is the signed integer represented by the concatenation of the high and low members of the time structure to form a 64-bit twos-complement binary number, with the high member as the most significant part and the low member as the least significant part.

If a time structure occurs as part of a value structure or as an element of an array pointed to by a value structure, its position number determines the value represented by the value structure, as described in 22.2.8. Otherwise, the time structure represents a value of type `TIME` defined in package `STANDARD`. The value is the product of the position number of the time structure and the resolution limit of the tool.

Bugzilla issue #238

Summary: What is `vhpiBootstrapFctT` for in `vhpi_user.h`?

Category: Technical

Description

C file `vhpi_user.h` contains `vhpiBootstrapFctT`. This isn't mentioned anywhere else in the text. Is this part of the standard and, if so, what are its semantics?

Resolution

In `vhpi_user.h`, changed definition to:

```
/****** Typedef for VHPI registration functions
******/
```

```
typedef void (*vhpiRegistrationFctT)();
```

In 20.2.1, page 352 (D4.2 clean), added the following to the end of the paragraph at line 3-4:

Each registration function shall be of the type `vhpiRegistrationFctT` defined in Annex B.

Bugzilla issue #242

Summary: Simple typo in D4.2

Category: Editorial

Description

Clean version page 207 line 60: "genertic" should be "generic"

Clean version page 524 line 53: "4" should be "3"

The referenced subtype index constraint (from 524 line 47) is (3 downto 0), so "4" is out of range; "3" matches the corrected version at 525 line 1.

Resolution

Fixed

Bugzilla issue #243

Summary: PSL-specific reserved words

Category:

Description

Should PSL-specific reserved words have some sort of additional mention or explanation in Clause 15.10 "Reserved words"?

For some members of this class of reserved word, e.g. "vmode" and "vprop", their appearance in the reserved words list is the only occurrence of the string in the document. Other words (e.g. "fairness" and "restrict") otherwise occur only in the VHDL LRM prose and not as a reference to the reserved word. For a VHDL user who does not use PSL, the existence of these words is unexplained by the LRM.

Additionally, the phrase "PSL keyword" occurs four times in the LRM and twice in the index, but is otherwise undefined.

I know PSL's definition is covered by a separate document and all definition or discussion of its keywords rightly belongs in that document. I think that the VHDL LRM would be clearer in this respect, though, if the PSL-specific reserved words were denoted in some fashion and identified as PSL keywords.

Resolution

See "Ballot comment #13" on page 10

Bugzilla issue #244

Summary: Ballot response: Make VHPI optional

Category: Technical

Description

In a negative ballot response William Hanna stated:

"Move VHPI to an Appendix -- Annex and state that it is optional."

Resolution

See "Ballot comment #4" on page 4

Bugzilla issue #245

Summary: References in both normative section and bibliography

Category: Editorial

Description

Jennie M Steinhagen reports that "It appears that IEEE Std 754 is in both the norm refs and the biblio. It should only be in one or the other."

Resolution

See "Ballot comment #3" on page 3

Bugzilla issue #246

Summary: Typos

Category: Editorial

Description

The following are typos in the clean version of D4.2. I have also included them in my ballot comments.

P63 L54: "The sequence of characters of abstract literal.." should read "The sequence of characters of the abstract literal..."

P77 L58: "...monitors and checkers, it not feasible..." should read "...monitors and checkers, it is not feasible..."

P88 L20-27: Spacing is applied inconsistently

P115 L2: "For such an attribute that denotes a function,..." should read ""For an attribute that denotes a function,..." "such" seems to be a leftover from an earlier version of the sentence.

P122 L55: "The unary logical operators belongs..." should read "The unary logical operators belong..."

P237 L64: "...by concatenating the to left of the expanded bit..." should read "...by concatenating to left of the expanded bit..."

P241 L9: "The reserved word range is also used as the name of a predefined attribute." So is "sub-type".

Resolution

See the following ballot comments:

“Ballot comment #6” on page 5

“Ballot comment #7” on page 6

“Ballot comment #8” on page 7

“Ballot comment #9” on page 7

“Ballot comment #11” on page 8

“Ballot comment #12” on page 9

“Ballot comment #14” on page 10

Bugzilla issue #247

Summary: Use of "bit" vs. "element"

Category: Technical

Description

The text on lines 17-51 on page 122 of the clean version uses both "bits" and "elements" to refer to the elements of R (e.g. "is the logical and of the bits of R", "the leftmost element of R"), but "bits" is never defined. I propose to replace "bits" by "elements" in these definitions. Alternatively, a definition for "bit" should be provided.

Resolution

See "Ballot comment #10" on page 8

Bugzilla issue #248

Summary: Wrong case for TRUE

Category: Editorial

Description

Annotated version p195, section 10.2 line 47:

Comment:

that wait until True;

Proposed Change:

I'm not sure, but I think it should be

that wait until TRUE;

Resolution

See "Ballot comment #5" on page 5

Bugzilla issue #249

Summary: Ballot comments from Alex Zamfirescu

Category: Technical

Description

IN P1076-2006-D4.2.pdf , USE TEXTUAL PAGE NUMBERS

<AZ1> Page 32, line 42 "signal GUARD or other" should be "signal GUARD and other"</AZ1>

<AZ2> Page 53, line 54 "conforms either to IEEE Std 754?-1985 [B2]56 or to IEEE Std 854?-1987 [B3]" newer revisions of these standards are available (check also other places)</AZ2>

<AZ3> Page 54, lines 52, 53 "of type BOOLEAN is TRUE if and only if the expression "S'EVENT and S" is TRUE, and FALSE otherwise" --if and only if-- does not take an --otherwise for Boolean--</AZ3>

Resolution

- re AZ1: Agreed. Changed as proposed.

- re AZ2: The latest revisions are as stated. The standards have been reaffirmed, but that does not change their approval year. The year of reaffirmation is listed in Clause 2, Normative references.

- re AZ3: Agreed. The text "and only if" is deleted.

Bugzilla issue #250

Summary: Extension of generate statements makes language too complex

Category: Technical

Description

Ballot comment from Z Navabi:

Extension of certain constructs like the generate statement to include case makes the language unnecessarily complex without much use for describing actual hardware.

Resolution

See "Ballot comment #15" on page 11

Bugzilla issue #251

Summary: New introduction needed

Category: Editorial

Description

Ballot comment from Stephen Schwarm:

"new introduction" needs some content. I suspect that is was a place holder

Proposed resolution:

I think much of what was in the original text was good and just needs editing.

Resolution

See "Ballot comment #16" on page 12

Bugzilla issue #252

Summary: Typo in 24.1.1

Category: Editorial

Description

Ballot comment by Steven Dovich:

The phrase "This allow an author of a VHDL description" reads badly.

Proposed resolution:

Change to "This allows and author of a VHDL description"

Resolution

“Ballot comment #17” on page 13

Bugzilla issue #253

Summary: Confusing terminology in 24.1.1

Category: Technical

Description

Ballot comment from Steven Dovich:

Inconsistency in terminology use leaves the text confusing. Specifically the use of the term decryption where description would be correct.

Proposed resolution:

Change the sentence:

"An encryption envelope is a portion of the description that is to be encrypted, and a decryption envelope is a portion of the decryption containing previously encrypted text to be decrypted."

to read:

"An encryption envelope is a portion of the description that is to be encrypted, and a decryption envelope is a portion of the description containing previously encrypted text to be decrypted."

Resolution

See “Ballot comment #18” on page 13