LCS 26:

VHDL Issue Number:	N/A
Classification:	Language Ambiguity Problem
Language Version:	VHDL-93 (1076a)
Summary:	Operator symbol for function within a protected type declaration or body.
Related Issues:	
Relevant LRM Sections:	
Key Words and Phrases:	shared variable, operator symbol
Current Status:	Submitted
1076-1993 Disposition:	N/A
Disposition Rationale:	N/A
Superseded By:	N/A

Date Submitted:	1998/09/11
Author of Submission:	John Willis (Based on VASG Discussion)
Author's Affiliation:	FTL Systems, Inc.
Author's Post Address:	1620 Greenview Drive SW, Rochester, MN 55902
Author's Phone Number:	1-507-288-3154
Author's Fax Number:	1-507-289-1108
Author's Net Address:	jwillis@ftlsys.com

Date Analyzed:	TBD
Author of Analysis:	TBD
Revision Number:	$Revision: 1.0 $
Date Last Revised:	$Date: 1998/10/29 11:30:17 $

Description of Problem

IEEE Standard 1076a allows function declarations in protected type declaration for which the declarator is an operator symbol, as in the example:

function "=" (i: integer; j: integer) return boolean;

Currently the invocation of such functions, using a prefix which is an object or protected type (implicit operand) and two explicit operands denoted in the call's association list, is counter intuitive (user-defined equality operator is triadic). However such a function cannot be declared with one less parameter according to VHDL LRM Section 2.3.1:

> *"The subprogram specification of a unary operator must have a single parameter. The subprogram specification of a binary operator must have two parameters; for each use of this operator, the first parameter is associated with the left operand, and the second parameter is associated with the right operand."*

Analysis & Rationale

The above problem is to be resolved through a change to VHDL LRM Section 2.3.1 to create special cases wherein the implicit parameter is considered as a parameter to the call if the left

operand is of monitored type.

For example:

variable b: boolean := sv1.”=”(sv2);

It is important to note that functions in the monitor interface may be called internally using notation of the form: *“=”(sv1, sv2)* as well as using the operator notation such as: *sv1 “=” sv2*. However in all cases, considering the selected name prefix as one operand, all operators retain the intuitively obvious number of parameters (as noted in Section 7).