This work has been accepted for publication in the *Proceedings of VIUF Fall '97 Conference*. Copyright may be transferred without notice, after which this version will be superseded and may no longer be accessible.

Reuse Through Genericity in SUAVE

Peter J. Ashenden
Dept. Computer Science
The University of Adelaide, SA 5005
Australia
petera@cs.adelaide.edu.au

Philip A. Wilsey and Dale E. Martin
Dept. ECECS, PO Box 210030
University of Cincinnati
Cincinnati, OH 45221−0030, USA
phil.wilsey@uc.edu, dmartin@ececs.uc.edu

Abstract

VHDL currently has a limited form of genericity in which component and entity declarations can be parameterized with formal generic constants. SUAVE extends the genericity mechanism by allowing formal generics types and by allowing generics to be specified in the interfaces of subprograms and packages. The approach is based on the features of Ada-95. It allows units to be re-used in a much wider variety of contexts without modifying the original code. In this paper, we show that the genericity added by SUAVE enhances reuse across the spectrum of modeling, from high-level to gate level. In particular, the genericity extensions interact with the SUAVE extensions for object-oriented data modeling to significantly improve support for high-level behavioral modeling and for developing test-benches. We show that the genericity extensions integrate seamlessly with the existing language. Furthermore, the implementation burden is not large, and, since generic instantiation is performed at elaboration time, there is no performance penalty in simulation or synthesis.

1. Introduction

One important objective of object-oriented extensions to VHDL is to improve support for re-use of models through the mechanism of inheritance. However, focussing on this approach ignores the important, orthogonal mechanism of genericity, which supports re-use in a much wider variety of contexts. SUAVE extends VHDL by adding both object-oriented and genericity features. The object-oriented extensions are described in a companion paper [3]; this paper describes the genericity extensions.

VHDL currently has a limited form of genericity in component and entity declarations. A component or entity can have formal generic constants, with actual constant values being supplied at elaboration time when the unit is instantiated. Generic constants are typically used in two ways. First, they are used to constrain the index bounds of array ports of a unit. This allows the unit to be re-used in contexts where the bounds of actual signals associated with the ports vary. Second, they are used to specify timing and operational parameters of a unit. This allows the unit to be re-used in contexts where the timing or operational requirements vary.

SUAVE extends the genericity mechanism of VHDL by allowing types to specified as formal generics and by allowing generics to be specified in the interfaces of subprograms and packages. The approach is based on the features of Ada-95. Use of formal type generics allows units to be reused in contexts where data of different types is to be manipulated. For example, a multiplexer can be specified with a formal type generic for the type of the input and output data. This allows the multiplexer model to be reused as a bit, bit_vector, std_logic, std_logic_vector, integer, or user-defined-type multiplexer, without modifying the original model code. Use of generics in the interfaces of subprograms and packages allows definition of container abstract data types that can be reused to contain data of different types. For example, a generic package can be defined to represent and manipulate sequences of integer, time values, or test vectors for different devices under test, again without modifying the original package code.

This paper outlines the SUAVE extensions for genericity and illustrates their use through some examples. A more complete description can be found in the SUAVE report [4]. Most of the features added to VHDL are adapted from features in Ada-95 [7], and are included largely for the same reasons that they are included in Ada-95 [5]. We

* This work was partially supported by Wright Laboratory under USAF contract F33615−95−C−1638.

show that the genericity added by SUAVE enhances reuse across the spectrum of modeling, from high-level to gate level.