NOTICE OF CHANGE

METRIC

MIL-HDBK-62
NOTICE 1
26 September 2002

DEPARTMENT OF DEFENSE
HANDBOOK

DOCUMENTATION OF DIGITAL ELECTRONIC SYSTEMS WITH VHDL

TO ALL HOLDERS OF MIL-HDBK-62:

1. THE FOLLOWING PAGES OF MIL-HDBK-62 HAVE BEEN REVISED AND SUPERSEDE THE PAGES LISTED:

NEW PAGE	DATE	SUPERSEDED PAGE	DATE
1-4	26 September 2002	1-4	13 September 1996
1-5	13 September 1996	1-5	Reprinted without change
2-13	26 September 2002	2-13	13 September 1996
2-14	13 September 1996	2-14	Reprinted without change
2-23	13 September 1996	2-23	Reprinted without change
2-24	26 September 2002	2-24	13 September 1996
3-25	13 September 1996	3-25	Reprinted without change
4-1	26 September 2002	4-1	13 September 1996
4-2	26 September 2002	4-2	13 September 1996
4-3	13 September 1996	4-3	Reprinted without change
4-8	26 September 2002	4-8	13 September 1996
4-9	26 September 2002	4-9	13 September 1996
G-5	26 September 2002	G-5	13 September 1996
G-6	13 September 1996	G-6	Reprinted without change
ST-1	26 September 2002	ST-1	13 September 1996

2. RETAIN THIS NOTICE AND INSERT BEFORE TABLE OF CONTENTS.

3. Holders of MIL-HDBK-62 will verify that page changes and additions indicated above have been entered. This notice page will be retained as a check sheet. This issuance, together with appended pages, is a separate publication. Each notice is to be retained by stocking points until the standard is completely revised or canceled.

Custodians:
Army - CR
Navy - EC
Air Force - 11
DLA - DH

Preparing activity:
DLA – CC

Review activity:
Air Force – 19

(Project 5962-1932)

AMSC N/A

FSC 5962

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

products list is discussed. The required structure and contents of VHDL descriptions provided to the Government, as defined by the VHDL DID, are presented. In particular, the requirement for both structural and behavioral models of each component of an electronic subsystem is described. This chapter provides guidelines to be used to tailor the VHDL DID and discusses an example of a tailored VHDL DID. This chapter also contains required annotations for VHDL models.

Chapter 5 contains a description of the construction and use of behavioral VHDL models. Common techniques used to create behavioral VHDL models, specify the timing for behavioral models, and annotate behavioral models are described. Also discussed are the usefulness of behavioral models in top-down design and the simulation of models with mixed levels of abstraction.

Chapter 6 discusses the construction and use of structural VHDL models. Common techniques used to create structural VHDL models, including automatic synthesis and schematic capture, are described. Applications of structural models for hybrid model simulation, physical design, testability analysis, and annotation with layout and testability information are also described in this chapter.

The preparation of VHDL models for simulation is detailed in Chapter 7. The process of configuring a model from libraries of component descriptions is described. Techniques that support the interoperability of models are emphasized. In component libraries these models can be combined freely to provide hybrid structural and behavioral models of systems. The development of test benches and test vectors to check the correctness and completeness of the model rather than the development of test vectors to check the correctness of the component design is discussed. Also discussed are the use of parameterized timing models and the selection of timing options for simulation.

Chapter 8 discusses issues surrounding VHDL modeling of the test and diagnostic functions of digital electronic systems. This chapter describes measures of and techniques for testability and describes different levels of testability based on the IEEE 1149 hierarchy of testing interfaces. The use of behavioral modeling to verify that the test bus and test controller systems respond properly to error conditions detected by on-chip BIT without requiring gate-level implementation details is emphasized. The use of detailed structural models as the starting point for built-in test structure generation, such as boundary scan, is discussed. This chapter also emphasizes that detailed structural models are necessary for evaluation of many testability measures.

Chapter 9 describes the preparation of a VHDL model for delivery to the Government. The contents and organization of the files delivered to the Government, as specified in the VHDL DID, are described. The files that must be delivered include not only the VHDL source models but also test vectors, annotations, certain other external files, and documentation. Chapter 9 also includes recommendations for VHDL model style and recommendations for naming files and organizing libraries.

REFERENCES

1. MIL-HDBK-454M, *General Guidelines for Electronic Equipment*, 28 April 1995.
2. DI-EGDS-80811, *VHSIC Hardware Description Language (VHDL) Documentation*, 11 May 1989.
3. IEEE Std 1029.1-1992, *waveform and Vector exchange Specification (WAVES)*, IEEE Design Automation Standards Subcommittee, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1992.
4. IEEE Std 1149.1-1990, *IEEE Standard Test Access Port and Boundary Scan Architecture*, IEEE Standards Board, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, May 1990.
5. P. McHugh, “IEEE P1149.5 Module Test and Maintenance Bus”, *IEEE design and Test of Computers* (December 1992).
6. IEEE Std 1164-1993, *IEEE Standard Multivalue Logic System for VHDL Model Interoperability*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, May 1993.
7. *Very High-Speed Integrated Circuits Final Program Report*, VHSIC Program Office, Office of the Under Secretary of Defense for Acquisition, Washington DC, September 1990.
8. J. Hines, “Where VHDL Fits Within the CAD Environment”, *24th ACM*/IEEE Design Automation Conference Proceedings*, Miami Beach, FL, June 1987, The Institute of Electrical and Electronics Engineers, Inc., New York, NY.
9. ANSI/IEEE Std 1076-1987, *IEEE Standard VHDL Language Reference Manual*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, March 1988.
10. ANSI/IEEE Std 1076-1993, *IEEE Standard VHDL Language Reference Manual*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, April 1994.
11. MIL-STD-1840C, *Automated Interchange of Technical Information*, 26 June 1997.
12. K. Parker and S. Oresjo, “A Language for Describing Boundary Scan Devices”, *Proceedings of the IEEE International Test Conference*, Los Alamitos, CA 1990, The Institute of Electrical and Electronics Engineers, Inc., New York, NY.
13. V. Berman, “An Analysis of the VITAL Initiative”, *VHDL Boot Camp*, VHDL International Users’ Forum, San Jose, CA, October 1993, VHDL Interna-

*Association for Computing Machinery

tional Users' Forum, c/o Conference Management Services, Menlo Park, CA.

14. O. Levia and F. Abramson, "ASCI Sign-Off in VHDL", *VHDL Boot Camp*, VHDL International Users' Forum, San Jose, CA, October 1993, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

BIBLIOGRAPHY

J. R. Armstrong, *Chip-Level Modeling in VHDL*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.

D. Coelho, *The VHDL Handbook*, Kluwer Acedemic Publishers, Norwell, MA, 1989.

R. Lipsett, C. Schaefer, and C. Ussery, *VHDL: Hardware Description and Design*, Kluwer Acedemic Publishers, Norwell, MA, 1989.

D. Perry, *VHDL*, McGraw-Hill Book Co., Inc., New York, NY, 1991.

VHSIC Annual Report for 1986, AD-A191-027, VHSIC Program Office, Office of the Under Secretary of Defense for Acquisition, Washington, DC, December 1986.

VHSIC Annual Report for 1987, AD-A199-880, VHSIC Program Office, Office of the Under Secretary of Defense for Acquisition, Washington, DC, December 1987.

VHSIC Annual Report for 1988, AD-A223-725, VHSIC Program Office, Office of the Under Secretary of Defense for Acquisition, Washington, DC, December 1988.

2-4.2 THE USE OF HIERARCHY IN STRUCTURAL DESCRIPTIONS

Hierarchy is important in structural models as a means of conveying the logical or physical decomposition of the hardware. Subpar. 10.2.3 of the VHDL DID (Ref. 1) requires that the hierarchy of a structural model follow the hierarchical organization of the physical design. This organization is useful in several ways. A hierarchical structure that corresponds to the physical organization supports the design and acquisition of upgrades by identifying physical interfaces between components that can be developed separately, and it can document maintenance issues. For example, a physically oriented hierarchical model reflects the organization of the hardware into line replaceable modules (LRMs). Also a hierarchical structure that corresponds to the physical organization documents boundaries between different technologies. A good structural hierarchy reflects the composition of boards into an interconnected set of integrated circuits with specific layout and routing. This partitioning facilitates the use of appropriate CAD tools for the design of integrated circuits and the design of boards.

The interconnection of components in a structural model should represent the physical interconnections. For example, each data-carrying wire on the board should have a corresponding signal in the VHDL model. The relationship between signals and wires may not be one-to-one, e.g., a 16-bit bus, which contains 16 individual wires, may be represented by a single signal in the VHDL model. This correspondence is one way of checking the consistency of the model with the physical hardware.

The physical hierarchy for a military digital electronic system has several levels that should be represented in a structural model. For example, a specification of a military system written to conform with MIL-STD-961 (Ref. 12) partitions the system into segments and the segments into configuration items including hardware configuration items (HWCIs). The HWCIs are further partitioned into prime items and critical items. A structural model of a digital electronic system should be consistent with this partitioning.

Hardware block diagrams and schematic diagrams are graphical representations of hardware data flow. VHDL provides mechanisms to represent this same hardware data flow formally. When a hardware block diagram is used to provide graphical documentation for a VHDL structural model, the following guidelines should be observed to make the relationship between the VHDL model and the block diagram clear and unambiguous:

1. There should be a one-to-one correspondence between the blocks in the diagram and component instantiations in the VHDL model.
2. Block names should be directly translatable into VHDL component instances. Either InputBus or input_bus is acceptable. The VITAL specification recommends names that use capital letters to separate words rather than underscores.
3. There should be a one-to-one relationship between interconnections in the block diagram and signals in the VHDL source program.
4. If the interconnections in the block diagram are labeled, the labels should be directly translatable into VHDL signal names.
5. All signals referenced in a VHDL process should have a corresponding interconnect in the block diagram.

Guideline 1 requires distinct instance labels but allows components to be reused. For example, the edge detection processor described in subpar 2-3.3 reuses the adder component within all of the filters. Guideline 2 encourages the user to translate automatically graphical block names into instance labels. (The block names may contain blanks that are translated into underscores in the VHDL source program.) Guideline 3 encourages the user to implement multibit busses and interconnects as bit vectors or higher level data types. For example, the behavioral model of the edge detection processor uses the integer data type for its signals. In a structural model these signals are translated into bit vectors. The use of single signals is essential for the mixed level of abstraction models described in par. 2-5.

A number of commercial CAD tools have the capabilities to create schematic representations of VHDL structural models and to create VHDL structural models directly from the schematic representations of the CAD tool.

2-4.2.1 Hierarchical Decomposition Based on Physical Elements

During design the digital electronic system is partitioned into subsystems. At the top level the system as a whole is described. The next level is a partitioning of the system into subsystems. The structural model should follow the partitioning described for the system into HWCI as described in the Level A specifications (Ref. 12). A structural model should preserve the partitioning into HWCIs of the physical system because it is a standard unit for acquisition.

The structural model should also be consistent with the physical hardware at the level of the line-replaceable unit (LRU). LRU partitioning is significant for logistics and support because it represents the basic unit used to maintain the system in the field. Any changes in boundaries between LRUs can have a significant effect on logistics and support; therefore, the structural model should accurately represent those boundaries. Furthermore, LRUs are important boundaries of the system for diagnostics and testing purposes. Field maintenance personnel must be able to isolate faults to an individual LRU. Thus a structural model should be able to simulate built-in test (BIT) diagnostic capabilities and interfaces to external test equipment at the level of its LRUs.

Another level of partitioning that should be represented in a structural model is the board. Partitioning the structural model to correspond to the physical partitioning of the hardware into boards assists in the automatic placement and routing of boards and in the thermal and power analysis of the boards. Furthermore, delays between boards are likely to be

much greater than delays within a specific board; this difference must be represented. In some cases the boards may be LRUs, so the components in the LRU-level partitioning and the board-level partitioning may be the same.

Partitioning of the structural model should also correspond to the partitioning of a board design into multichip modules (MCMs) and integrated circuits (ICs) as appropriate. Different CAD tools and optimization criteria may apply at the MCM/IC level versus the board level, so partitioning of a structural model to represent MCMs can aid in the synthesis, analysis, and optimization of a design using MCMs.

Partitioning of the structural model should also correspond to the partitioning of an MCM design into packaged very large-scale integrated (VLSI) circuits. Because packaged VLSI circuits are the lowest possible practical level for repair through replacement, isolation of faults to specific integrated circuits is an important design consideration. Also, if the model accurately represents the boundaries of VLSI circuit packages, VLSI CAD tools can be used to synthesize, analyze, and optimize VLSI circuits.

Structural models for components of a circuit should also follow the partitioning used by the CAD tools to design the circuit. For example, the hierarchy of the structural model of a VLSI circuit should follow the boundaries of standard cells or macrocells used by the CAD tool. In general, if a CAD tool is used to design a circuit and generate a VHDL model automatically for the circuit, the generated description follows the hierarchy of the design. A CAD tool that flattens a design hierarchy before producing a structural model of the design should not be used to generate models for delivery to the Government. Using CAD tools to generate detailed and hierarchical structural models is a recommended practice since it reduces costs and helps to keep the model consistent with the physical hardware.

2-4.2.2 Leaf Modules in a Hierarchical Structural Description

If a component is represented by a behavioral model and does not have a structural model, the component is called a leaf module. Subpar. 10.2.1.1 of the VHDL DID (Ref. 1) specifies three valid leaf module options:

1. Modules selected from a Government list of valid uses of leaf modules referenced or contained in the contract.
2. Modules corresponding to a collection of hardware elements that together exhibit a stimulus-response behavior but whose interaction is best modeled at the electrical or physical level.
3. Modules whose detailed design has not yet been completed but whose behavior is required as a delivery disclosure at specified times during the contract.

The first option for a leaf module allows the contractor to use models from a Government source of validated models. The Government requires VHDL models for the electronic components delivered to it. These requirements are discussed in Chapter 4. Once these models have been validated, they can be supplied to contractors for use in VHDL models of hardware systems that use the products. The Government and the contractor may also negotiate to include other VHDL models, such as models not in the qualified products list (QPL) that are developed by the contractor or by other Government contractors. These negotiations must be reflected in the tailored VHDL DID for the specific contract.

The second option identifies a common set of primitive elements used in designs whose elements are not easily described accurately with VHDL behavioral models. As described in subpar. 10.2.1.1 of the VHDL DID (Ref. 1), these elements include digital logic gates, analog circuit blocks, and power supplies. Functional models of digital logic gates are defined as part of the IEEE Std 1164 (Ref. 2) specification of standard signal formats. This specification includes truth tabes and a resolution function for using a nine-value state/strength logic system for AND, OR, NOT, NOR, NAND, and XOR. This functional specification is being augmented with timing information and standard formats for back-annotation by the VITAL effort (Ref. 3).

The third option is designed to cover situations in which the Government wants VHDL models delivered during the design cycle, i.e., before design of all of the components has been completed. In this case high-level behavioral models may be used as leaf modules to specify the current state of the design. As the design progresses into more detail, these behavioral models are augmented with structural models.

2-4.3 EXAMPLES OF STRUCTURAL DESCRIPTIONS

In this subparagraph two examples of structural VHDL models are presented: one at algorithmic level and one at a register-transfer level. The algorithmic model uses the data-type definitions and some of the functions of the sobel_algorithm library presented in subpar. 2-3.3. The entity interface declarations and architecture bodies for this level of model are included in this library. The register-transfer-level model uses different data-type definitions, in which the number of bits in each word is specified. These definitions and the entity interfaces and architecture bodies that reference these packages are in the sobel_structure library.

2-4.3.1 Algorithmic-Level Structural Description

Fig. 2-8 shows a hierarchy for an algorithmic structural model of the edge detection system described in subpar. 2-3.3. This model is at the algorithmic level because the data types have not yet been refined to bit vectors; therefore, the inputs and outputs of the model are not bit-for-bit representations of the inputs and outputs of the real device. However, the structural model does reveal much of the physical organization of the system as it will be implemented. As shown in Fig. 2-8, this model continues to use some of the elements of the behavioral model, particularly the weight function, and it uses the data-type definitions previously used in the behavioral model. This structural model implements the

ly so that structural models can be interfaced at a later stage. In general, the structural VHDL models use low-level data types such as the IEEE standard logic types (Ref. 2) as the data types of their input and output ports. The behavioral model should be prepared to interface with such data types. In some cases a single behavioral input or output may correspond to an array of standard logic values.

One VHDL mechanism that supports interfacing behavioral and structural models is the type conversion function. Type conversion functions can be associated with the ports of structural models in either component instances or configuration specifications. In the early stages of model development, the project manager should develop a standard set of data types for the module interfaces. All models should be constructed with these standard data types. VHDL provides a mechanism (the package) to share a single definition of a data type across all parts of a model.

2-5.3 AN EXAMPLE OF A MIXED LEVEL OF ABSTRACTION MODEL

The hierarchy of a mixed level of abstraction model is shown in Fig. 2-19. This model uses the register-transfer-level behavioral structural models of the components of the horizontal filter processor in conjunction with ISA-level behavioral models of the vertical and diagonal processors and with algorithmic-level behavioral models of the memory processor and the magnitude and direction processor. Because integer formats are used in the behavioral models for the memory processor and the magnitude and direction processor, type conversion functions are required to convert the integers to and from the 8-bit array inputs and the 12-bit array outputs of the structural model of the horizontal processor.

REFERENCES

1. DI-EGDS-80811, *VHSIC Hardware Description Language (VHDL) Documentation*, Department of De-fense, Washington, DC, 11 May 1989.
2. IEEE Std 1164-1993, *IEEE Standard Multivalue Logic System for VHDL Model Interoperability (std_logic_1164)*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, May 1993.
3. IEEE Std 1076.4, *IEEE Standard for VITAL Application-Specific Integrated Circuit (ASIC) Modeling Specification*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, December 1995.
4. D. Gajski and R. Kuhn, “Guest Editors’ Introduction: New VLSI Tools”, IEEE Computer 16, 11-4 (1983).
5. G. A. Frank, C. U. Smith, and J. L. Cuadralo, “An Architecture Design and Assessment System for Software/Hardware Codesign”, *Proceedings of the 22nd Design Automation Conference*, Las Vegas, NV, June 1985, pp. 417-24, IEEE Computer Society Press, Los Alamitos, CA.
6. J. Schoen, *Performance and Fault Modeling with VHDL*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.
7. R. A. Walker and D. E. Thomas, “A Model of Design Representation and Synthesis”, *Proceedings of the 22nd Design Automation Conference*, Las Vegas, NV,

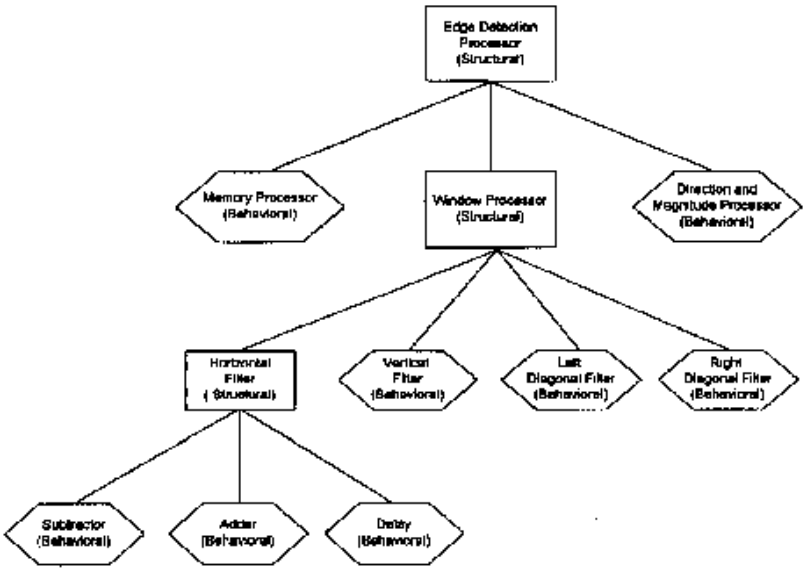

Figure 2-19. Hierarchical Organization of a Mixed Level of Abstraction Model

June 1985, pp. 4543-9, IEEE Computer Society Press, Los Alamitos, CA.

8. D. P. Siewiorek, C. G. Bell, and A. Newell, *Computer Structures: Principles and Examples*, McGraw-Hill Book Co., Inc., New York, NY, 1982.
9. J. R. Armstrong and F. G. Gray, *Structured Logic Design Using VHDL*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.
10. J. E. DeGroat and G. S. Powley, Jr., "Object-Oriented Generation of VHDL Synthesizable Architectural Building Blocks", *VHDL: Champions of the Second Generation, Proceedings VHDL International Users' Forum Spring Conference*, 2-6 April 1995, San Diego, CA, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.
11. C. O. Scheper and R. L. Baker, "VHSIC Silicon Compiling Tools Applied to Image Processing", *1987 Digest of Papers, Government Microcircuit Applications Conference*, GOMAC-87B119187, Orlando, FL, 26-29 October 1987, US Army Laboratory Command, Fort Monmouth NJ.
12. MIL-STD-961A, *Defense Specifications*, 22 March 1995.

BIBLIOGRAPHY

A. G. Stanculescu, A. S. Tsay, N. D. Zamfiresccu, and D. L. Perry, "Switch-Level VHDL Descriptions", *IEEE International Conference on Computer-Aided Design, Digest of Technical Papers*, 1989, pp. 180-3, IEEE Computer Society Press, Los Alamitos, CA.

J. R. Armstrong, "Accurate Timing Modeling in Behavioral Models", SIGDA Newsletter 18, 72-5 (December 1988).

J. R. Armstrong, *Chip-Level Modeling With VHDL*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

D. L. Barton, "Behavioral Descriptions in VHDL", VLSI Systems Design 9, 28-33 (June 1988).

M. J. Chung and S. Kim, "An Object-Oriented VHDL Design Environment", *Proceedings of the 27th ACM/IEEE Design Automation Conference*, Piscataway, NJ, 1991, Institute of Electrical and Electronic Engineers, Inc., New York, NY.

A. Dewey, "The VHSIC Hardware Description Language (VHDL) Program", *ACM IEEE 21st Design Automation Conference Proceedings 84*, Albuquerque, NM, 23-27 June 1984, Institute of Electrical and Electronic Engineers, Inc., New York, NY.

R. Ernst and J Bhasker, "Simulation-Based Verification for High-Level Synthesis", IEEE Design & Test of Computers 8, 14-20 (March 1991).

F. T. Hady, J. H. Aylor, R. D. Williams, and R. Waxman, "Uninterpreted Modeling Using the VHSIC Hardware Description Language (VHDL)", *1989 IEEE International Conference on Computer-Aided Design, Digest of Technical Papers*, 1989, IEEE Computer Society Press, Los Alamitos, CA.

P. Hunter, R. Harr, and H. Carter, "General Requirements for VHDL Behavioral Models", SIGDA Newsletter 18, 21-22 (December 1988).

M. Neighbors, "Extending the VHDL Simulation Environment into the Requirements Analysis Level – An Example (SINCGARS-ICOM Radio)", *Proceedings of the Tactical Communications Conference, Vol. 1, Tactical Communications, Challenges of the 1990s*, Fort Wayne, IN, 24-26 April 1990, pp. 435-53, Institute of Electrical and Electronics Engineers, Inc., New York, NY.

N. Park and A. Parker, "A Software Package for Synthesis of Pipelines From Behavioral Specifications", IEEE Transactions on Computer-Aided Design 7 (March 1988).

R. A. MacDonald and R. Waxman, "Operational Specifications of the SINCGARS Radio in VHDL", *Proceedings of the Tactical Communications Conference, Vol. 1, Tactical Communications, Challenges of the 1990s*, Piscataway, NJ, 1990, pp.415-33, Institute of Electrical and Electronics Engineers, Inc., New York, NY.

R. J. Hookway, "System Simulation Using VHDL", *Automated Design and Engineering for Electronics – West, Proceedings of the Technical Sessions*, 31 March – 2 April 1987, pp.21-33, Cahners Exposition Group, Anaheim, CA.

D. W. Runner and E. H. Warshawsky, "Synthesizing Ada's Ideal Machine Mate", VLSI Systems Design 9, 30, 32-3, 36, 38-9 (October 1988).

B. R. Stanisic, "VHDL Modeling for Analog-Digital Hardware Designs (VHSIC Hardware Description Language)", *IEEE International Conference on Computer-Aided Design, Digest of Technical Papers*, 1989, pp. 184-7, IEEE Computer Society Press, Los Alamitos, CA.

S. Tandri, M. H. Abd-El-Barr and C. McCrosky, "High-Level Specification, Simulation, and Layout Generation of Systolic Arrays – A Case Study", *Conference Proceedings, CCVLSI '90, Canadian Conference on Very Large-Scale Integration*, pp. 6.6/1-6, Carlton University, Ottawa, Ontario, Canada, 1990.

A. Sama and J. Armstrong, "Behavioral Modeling of RF Systems With VHDL", *Proceedings of the Spring 1991 VHDL Users' Group Meeting*, 1991, Cincinnati, OH, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

D. Coelho, *The VHDL Handbook*, Kluwer Academic Publishers, Norwell, MA, 1989.

B. Cohen, *VHDL Coding Styles and Methodologies: An In-Depth Tutorial*, Kluwer Academic Publishers, Norwell, MA, 1995.

A. Dewey, *Analysis and Design of Digital Systems With VHDL*, Addison-Wesley Publishing Company, Inc., Piscataway, NJ, 1992.

A. Dewey, "The VHSIC Hardware Description Language (VHDL) Program", *ACM/IEEE 21st Design Automation Conference Proceedings 84*, Piscataway, NJ, 1984, Association of Computing Machinery, Baltimore, MD.

Enabling Design Creativity, Proceedings of the VHDL International Users' Forum Fall 1991 Meeting, Newport Beach, CA, October 1991, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

R. Harr and A. Stancluescu, Eds., *Applications of VHDL to Circuit Design*, Kluwer Academic Publishers, Norwell, MA, 1989.

S. Leung and M. A. Shanblatt, *ASIC System Design With VHDL: A Paradigm*, Kluwer Academic Publishers, Norwell, MA, 1989.

R. Lipsett, C. Schaefer, and C. Ussery, *VHDL: Hardware Description and Design*, Kluwer Academic Publishers, Norwell, MA, 1989.

Z. Navabi, *VHDL: Analysis and Modeling of Digital Systems*, McGraw-Hill Book Co., Inc., New York, NY, 1993.

D. Perry, *VHDL*, McGraw-Hill Book Co., Inc., New York, NY, 1991.

J. Schoen, *Performance and Fault Modeling With VHDL*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

Standard No. 1076-CONC-1990, *The Sense of VASG*, 1990. (This publication is the companion document to IEEE Std 1076-1987.)

Using VHDL for Electronic Product Design, Proceedings of the VHDL Users' Group Spring 1991 Meeting, Cincinnati, OH, April 1991, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

Using VHDL in System Design, Test, and Manufacturing, Proceedings of the Spring 1992 VHDL International Users' Forum, Scottsdale, AZ, 1992, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

VHDL Boot Camp, Proceedings of the Fall 1993 VHDL International Users' Forum, San Jose, CA, October 1993, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

VHDL: Windows of Opportunity, Proceedings of the VHDL Users' Group Fall 1990 Meeting, Oakland, CA, October 1990, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

REPRINTED WITHOUT CHANGE.

CHAPTER 4
DoD REQUIREMENTS FOR THE USE OF VHDL

In this chapter the two primary Government documents concerning the use of VHDL are discussed: (1) MIL-HDBK-454 and (2) the VHDL DID, DI-EGDS-80811. The need for VHDL descriptions of all application-specific integrated circuits (ASICs) and all qualified digital electronic integrated circuits in board-level designs is dis-cussed. The DID-required structure and contents of VHDL descriptions provided to the Government are presented. In particular, the requirement for both structural and behavioral models of each component of a digital electronic subsystem is described. This chapter also describes the required annotations to VHDL models.

4-1 INTRODUCTION

The two primary documents that describe the requirements of very high-speed integrated circuit (VHSIC) hard-ware description language (VHDL) models to be delivered to the Government are MIL-HDBK-454 (Ref. 1) and the VHDL Data Item Description (DID), DI-EGDS-80811 (Ref. 2). MIL-HDBK-454 describes the criteria for selection and application of various types of electronic equipment. In particular, Guideline 64 of MIL-HDBK-454 describes such criteria for microelectronic devices and provides guidance to deliver VHDL models of application-specific integrated circuits (ASIC) and microelectronic circuits used in board designs. Further, these models should comply with requirements stated in the VHDL DID. The VHDL DID lays out comprehensive requirements for VHDL models and the necessary auxiliary and testing support files.

VHDL is also required by MIL-STD-1840 (Ref. 3) for the exchange of digital data relating to electrical or electronic applications. MIL-STD-1840 requires one or more of the following formats:

1. Electronic Design Interchange Format (EDIF) (Ref. 4)
2. VHDL (Ref. 5)
3. International Graphics Exchange Standard (IGES) (MIL-PRF-28000) (Ref. 6)
4. Institute for Interconnecting and Packaging Electronic Circuits (IPC) (Ref. 7)

Subpar. 4.4.11.2 of MIL-STD-1840 cites the VHDL DID (Ref. 2) and Electronic Industries Alliance standard EIA-567 (Ref. 8) as the application protocols used to organize and write the VHDL code. Though MIL-HDBK-454, the VHDL DID, and MIL-STD-1840 require the use of VHDL, they provide little or no practical guidance on the organization of VHDL models and support files.

This chapter contains approaches to structuring the VHDL models so that DID requirements and intent can be met with appropriate auxiliary and testing support files. These approaches are written to readily support the tailoring of items in the DID to fit project requirements and the structuring of VHDL models so that they can be delivered to the Government at an affordable cost.

4-2 MIL-HDBK-454 GUIDELINES FOR THE USE OF VHDL

MIL-HDBK-454 describes the common guidelines to be used in military specifications for electronic equipment. It contains 78 individual guidelines covering a variety of issues relating to electronic equipment. Guideline 64 of MIL-HDBK-454 (Ref. 1) covers microelectronic devices and recommends delivery of VHDL models for microelectronic circuits under specific situations. Microelectronic circuits include monolithic integrated circuits, hybrid integrated circuits, and multichip modules.

Subpar. 4.1.3 of Guideline 64 of MIL-HDBK-454 (Ref. 1) describes a sequence of choices to be used to acquire microelectronic circuits. Subpar. 4.5.1 of Guideline 64 recommends delivery of structural and behavioral models for ASICs and cites the VHDL DID (Ref. 2). Otherwise, a nonstandard part approval must be requested. MIL-HDBK-454 lays out the requirements for the documentation and testing of nonstandard and standard parts on the Qualified Products List and of other microcircuits.

Subpar. 4.5.3 of Guideline 64 recommends documentation of digital qualified devices used in board-level applications with behavioral VHDL descriptions. These behavioral descriptions must enable test generation and support fault detection/isolation to the circuit pins.

4-2.1 DOCUMENTATION OF ASICs DEVELOPED FOR THE GOVERNMENT WITH VHDL

One form of a nonstandard microelectronic circuit used increasingly in military electronic systems is an ASIC. An ASIC is any microcircuit customized to perform a specific function. By dedicating all resources on the device to a specific function, ASICs provide high throughput for a given level of power, weight, and size. The rapidly increasing capability of electronic computer-aided design (ECAD) tools has made it possible to design and fabricate ASICs at a reasonable cost. However, the small number of copies of ASICs makes them especially vulnerable to becoming unavailable due to a lack of production facilities. The existence of both behavioral and structural VHDL models for ASICs means

that ECAD capabilities can be used either to reengineer the function of the ASIC for new fabrication technologies or to transfer the design automatically to a new manufacturer's production line.

Subpar. 4.5.1 of MIL-HDBK-454 (Ref. 1) recommends that the circuit design of digital microelectronic ASICs developed be documented with behavioral and structural VHDL descriptions. The behavioral VHDL description must model both the function and timing of the microcircuit at the ports of the model. The behavioral VHDL model must be sufficiently detailed to permit its use within a larger VHDL model for test generation and fault grading of the larger model.

Subpar. 4.5.4 of MIL-HDBK-454 (Ref. 1) recommends that the test vectors and test waveforms for digital ASICs be documented and delivered to the Government in Waveform and Vector Exchange Specification (WAVES) format.

In an information section, par. 5.6 of MIL-HDBK-454 (Ref. 1) references the VHDL DID (Ref. 2) as a guideline to be used to prepare the VHDL documentation of an ASIC.

Although MIL-HDBK-454 does not provide specific guidance on structural models, the information can be inferred from the VHDL DID (Ref. 2). As a guideline, the structural model of digital microcircuits should be sufficiently detailed to support fault coverage analysis based on the equivalence classes of single, permanent, stuck-at-zero, and stuck-at-one faults on all lines (i.e., interconnects). In general, this requirement implies a structural model that is decomposed into gate-level primitive modules and atomic storage functions, such as flip-flops. However, large regular structures, such as read-only memories (ROMs) and random-access memories (RAMs), can be treated as atomic structures provided they are tested using the appropriate algorithms.

Subpar. 4.5.4 of Guideline 64 of MIL-HDBK-454 recommends that the ASIC test stimuli be written and documented in Waveform and Vector Exchange Specification (WAVES) (Ref. 9).

Chapter 7 describes the WAVES standard and how to implement a VHDL test bench using WAVES.

4-2.2 DOCUMENTATION OF QUALIFIED DIGITAL INTEGRATED CIRCUITS WITH VHDL

Subpar. 4.5.3 of Guideline 64 of MIL-HDBK-454 recommends documentation with VHDL descriptions of microelectronic circuits used in board-level designs. These descriptions must fully define the functions of the device and must include timing of the device at the input/output (I/O) ports in sufficient detail to support test generation, fault detection, and fault isolation to the device when board or subsystem simulation is performed. The behavioral VHDL model recommended by MIL-HDBK-454 should be suitable for use as a leaf module in a VHDL model of a system using the modeled device.

4-2.3 THE LIBRARY OF VHDL DESCRIPTIONS OF STANDARD DIGITAL PARTS

Under the auspices of the Defense Supply Center, Columbus (DSCC), the Department of Defense (DoD) has started building a library of interoperable VHDL descriptions of microelectronics circuits. This VHDL model library (VML) acts as a standardization vehicle and is available to Government contractors to enable them to design systems quickly. As a result, the DoD will receive more VHDL designs for future use. Validated models placed in the VML will be a resource to aid design engineers in system upgrades or to provide logistical support after system delivery. VML information may be obtained by emailing either the VHDL or ASIC Group at: ASICS@dscc.dla.mil.

DoD project managers who are receiving VHDL models should contact DSCC to alert them to the existence of the models, they should work with DSCC on the specification and validation of these models, and they should send a copy of the VHDL models to DSCC. DoD project managers who are tailoring the VHDL DID and defining acceptable leaf-level modules should contact DSCC to find out whether the VML has VHDL models that could be used by the program. The VHDL DID (Ref. 2) requires Government approval of leaf-level models used in higher level VHDL models. In the future the VML may provide a source for such leaf-level models.

The models in such a library must have sufficient accuracy and quality to allow their use as formal models for parts of Government-procured systems. To deal with this issue, the US Air Force has published a procedure for validating VHDL models (Ref. 10). DSCC is evaluating these validation techniques in order to use them to screen models delivered to the DSCC VHDL model library. The validation process traces its requirements back to MIL-HDBK-454 and to the VHDL DID.

4-2.4 TEST BENCH REQUIREMENTS FOR VHDL DESCRIPTIONS

VHDL has popularized the concept of a test bench, a collection of VHDL modules that apply stimuli to a module under test (MUT). Test benches may also compare the response of the MUT with the expected output and report any differences between observed and expected responses. WAVES provides mechanisms for generating VHDL test benches and for using a standard format for the external files. WAVES is described in more detail in Chapter 7.

4-3 OVERVIEW OF THE VHDL DATA ITEM DESCRIPTION

The VHDL Data Item Description, DI-EGDS-80811 (Ref. 2), provides a definition of the Department of Defense requirements for a delivered VHDL model. The DID can be tailored for particular contracts to meet the unique requirements of a specified program. This tailoring specifies the

models to be developed and delivered, and it may further define some of the basic terms used in the DID, such as "stand-alone-modules". The DID can be tailored by rewriting its sections.

Appendix B contains an example of a tailored VHDL DID, including both the text of the initial DID and the changes that were made to it.

The VHDL DID requires delivery of a hierarchy of VHDL module descriptions. This hierarchy must be consistent with the hierarchy of the physical hardware (Ref. 2, sub-par. 10.2.1), as described in Chapter 2. A VHDL module is defined by the DID as a deliverable item that includes several files and VHDL design units. The DID requires one VHDL module to be defined for the entire system and one for each physical electronic unit, such as an assembly, sub-assembly, or integrated circuit. VHDL modules should also be defined for important subsections or groupings of complex physical units. For each VHDL module of the design, the VHDL DID requires an associated VHDL entity interface, one or more behavioral bodies, and (except for leaf modules) a structural body. Furthermore, the VHDL DID requires a VHDL test bench for each stand-alone module.

An important aspect of tailoring the VHDL DID to a specific project is specifying the hierarchy of VHDL modules that will be delivered. Each of these VHDL modules requires its own test bench and its own structural and behavioral models. Within the VHDL modules the contractor is encouraged to use VHDL hierarchies to clarify design.

Subpar. 10.2.2.3 of the VHDL DID requires that operating conditions for the physical hardware module be characterized in the corresponding VHDL entity interface. Operating characteristics include temperature range, logic level definitions (which relate the logic values used in the simulation to voltage levels in the physical design), power and heat dissipation, and radiation hardness. The VHDL DID also requires that VHDL packages be used to encapsulate this information when it can be reused across multiple VHDL modules. This use of packages is consistent with standards such as Institute of Electrical and Electronics Engineers (IEEE) 1164 (Ref. 11), WAVES (IEEE 1029.1) (Ref. 9), EIA-567 (Ref. 8), and VITAL (Ref. 12). One area of tailoring of the DID relates to the use of these standards. MIL-HDBK-454 (Ref. 1) specifies the use of WAVES. The computer-aided acquisition and logistics support (CALS) standard, MIL-STD-1840 (Ref. 3), specifies the use of EIA-567. A tailored DID can refer to other standards, such as IEEE 1164 (Ref. 11), IEEE 1149.1 (Ref. 13), and VITAL (Ref. 12).

Development of models without the use of standards runs the risk of reducing the interoperability and reuse potential of the models. Requiring the use of standards after model development has begun is very expensive. Also model developers should use standard packages so that models will work together when they are integrated.

4-3.1 ENTITY INTERFACE REQUIREMENTS

Subpar. 10.2.2 of the VHDL DID (Ref. 2) defines the requirements for the declaration of design entities as follows: "The entity declaration shall include an interface declaration which describes the input and output ports of the system. The entity declaration shall also describe timing and electrical requirements for the behavior of the device and allowable operating conditions. The entity declaration shall also include explanatory comments."

These comments should identify the corporate and individual authors of the entity interface, the date and time of the last revision of the design interface, and identification of the device being modeled. The entity interface declaration should include references to VHDL libraries and packages that are required by every body for the interface. Libraries and packages specific to a particular architecture should not be included.

The entity interface can also include assertions about the interface, including relationships between the input and output ports and conditions on the value and timing of the entry and exit of input and output data. Assertions should be used to describe requirements on the module, and the timing delays in the behavioral bodies should capture the actual behavior of the physical device. If a behavioral body is used to describe a design for which no corresponding physical hardware exists, the behavioral body must be clearly commented to indicate the source of the timings.

4-3.1.1 Entity Names

Subpar. 10.2.2.4 of the VHDL DID (Ref. 2) requires that names for the VHDL entities be traceable to the names of the corresponding physical electronic components whenever such a correspondence exists. Similarly, the names of architecture bodies for a design entity should reflect a distinguishing implementation characteristic of that architecture body, such as the level of abstraction, the technology used to implement the component, or the manufacturer. This traceability is important for verification that the model is complete, i.e., each physical hardware component is instantiated in the VHDL design. Appropriate naming also aids verification that the VHDL model design hierarchy is consistent with the physical design hierarchy. Appropriate names for entity interfaces also aid maintenance of the model, particularly when upgrades are made to physical components. A well-structured VHDL model of a system allows changes to be isolated to those parts of the model that correspond to the physical components being upgraded and perhaps to configurations of those components.

4-3.1.2 Input and Output Definitions

Subpar. 10.2.2.1of the VHDL DID (Ref. 2) required that each entity interface shall describe all input and output ports. In particular for very large-scale integrated (VLSI) circuits, there should be a port declared for each pin of the circuit. This requirement is driven by the needs of WAVES (Ref. 9).

delivery to the Government need to be identified in this list. For the first delivery of a model to the Government, this file should be empty.

The sixth file is a list of VHDL modules that originate with this delivery. For example, if this is the first time that a model is being delivered to the Government, all the VHDL modules are listed in this file. If since the last delivery of the model to the Government, the model hierarchy has been extended to include more detail, this file will identify the new VHDL modules. For example, if a behavioral model of a component has been augmented with a structural model that references several new behavioral models, these structural and behavioral models will be referenced in this list as will the new structural model.

The seventh file associates VHDL modules with their corresponding test benches. For each VHDL module there is a list of corresponding test benches. For each test bench there is a list of VHDL entities comprising that test bench. This list includes not only the VHDL entity for the MUT but also the all of the VHDL entities that are components of the MUT and all of the VHDL entities that are part of the test bench external to the MUT.

The files after the seventh specified file contain VHDL design units and auxiliary files. The auxiliary files proceed VHDL design units. Auxiliary files include WAVES header files, WAVES external files, timing files (such as standard delay format (SDF) files used by VITAL (Ref. 12)), and external environment parameter files (such as data files for behavioral models). VHDL MUT descriptions must be distinguished from VHDL test bench descriptions.

The delivery medium is another place where tailoring of the VHDL DID is important. Par. 7.3 of the VHDL DID defines the preferred media as nine-track magnetic tape, 1600 bits per inch, unlabeled, with 80-character records and 24 records per block. An identifying label must be attached to the tape reel, and a hardcopy of Files 1 and 2 must be included with the tape. Because of the wide variety of computer systems in existence, the Government may want to specify other magnetic media or other delivery format.

4-3.7 REQUIRED ANNOTATIONS OF VHDL MODULES

The VHDL DID requires explanatory comments to make the intent of a VHDL module clear. The following information is required:

1. Any factors restricting the general use of the VHDL module to represent the modeled hardware. For example, if nonstandard signal state/strength definitions are used, they should be noted in the explanatory comments.
2. General approaches taken to modeling, particularly decisions regarding model fidelity. Model fidelity information includes information about the timing models used and any variance in exact function from the subject hardware (such as the use of host-dependent floating point formats for calculations).
3. Any further information the originating organization considers vital to subsequent users of the descriptions. For example, if the source code for the VHDL module has been structured to support a particular synthesis tool, this fact should be noted with the version of the tool.

VHDL modules selected from the list of Government-approved modules and VHDL modules that have been previously accepted by the Government require documentation of the originator or source of the VHDL model, a DoD-approved identifier is such an identifier exists, and a design unit name or revision identifier.

A revision history must be maintained for design units that have been accepted by the Government. The design revision history is included as comments in the design unit. The revision history must include the dates of revisions, the performing individual and organization, the rationale for the revision, a description of what part of the design unit required revision, and the testing done to validate the revised model. Revision histories should also be maintained for auxiliary files using the same format as the VHDL design units where possible.

4-3.8 AN EXAMPLE OF A TAILORED DID

Appendix B, “Example of a Tailored DID”, includes an example of a DD Form 1423 used to tailor the VHDL DID. This form describes a contract data requirement. Seven remarks are listed that specify the deliverables for the VHDL DID and reference the VHDL DID paragraph numbers. This modified form of the DID specifies a series of six versions of the model to be delivered. (These versions are called “levels” in DD Form 1423.)

Four behavioral model versions are required: an architectural level model, two application level models, and a bus functional model, which is a mixed abstraction level model.

Two structural model versions are required: a structural model whose leaf-level entities are integrated circuits and a structural model at the register transfer level of abstraction.

The tailoring also requires that the models of input stimuli and output results be specified in IEEE Std 1029.1 format (WAVES).

REFERENCES

1. MIL-HDBK-454M, *General Requirements for Electronic Equipment*, 28 April 1995.
2. DI-EGDS-80811, *VHSIC Hardware Description Language (VHDL) Documentation*, Department of Defense, Washington, DC, 11 May 1989.
3. MIL-STD-1840C, *Automated Interchange of Technical Information*, 1997.
4. EIA-548, *Electronic Design Interchange Format (EDIF)*, Electronic Industries Alliance, Arlington, VA, 1989.
5. IEEE Std 1076-1993, *IEEE Standard VHDL Language Reference Manual*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, April 1994.

6. MIL-PRF-28000B, *Digital Representation for Communication of Product Data: IGES Application Subsets and IGES Application Protocols*, 30 September 1999.
7. IPC-D-351, *Printed Board Drawings in Digital Form*, Institute for Interconnecting and Packaging Electronic Circuits (IPC), 1989.
8. EIA-567-A, *VHDL Hardware Component Modeling and Interface Standard*, Electronic Industries Alliance, Arlington, VA, March 1994.
9. IEEE Std 1029.1-1991, *Waveform and Vector Exchange Specification*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1991.
10. *VHDL Model Verification and Acceptance Procedure*, Rome Laboratories/ERDD, Department of the Air Force, Griffiss Air Force Base, Rome, NY, March 1992.
11. IEEE Std 1164-1993, *IEEE Standard Multivalue Logic System for VHDL Model Interoperability*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, May 1993.
12. IEEE Std 1076.4-1995, *IEEE Standard for VITAL Application-Specific Integrated Circuits (ASIC)*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1996.
13. IEEE Std 1149.1-1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, May 1990.
14. J. Hallenbeck, J. Cybrynski, N. Kanopoulos, T. Markas, and N. Vasanthavada, *The Test Engineer's Assistant, A Support environment for Hardware Design for Testability*, IEEE Computer Society Press, Los Alamitos, CA, April 1989.

BIBLIOGRAPHY

MIL-H-38534B, *General Specification for Hybrid Microcircuits*, 7 July 1993.
MIL-STD-883D, *Test Methods and Procedures for Microelectronics*, 15 November 1991.
MIL-M-38510J, *General Specification for Microcircuits*, 15 November 1991.
MIL-I-38535B, *General Specification for Integrated Circuits (Microcircuits) Manufacturing*, 1 June 1993.
MIL-HDBK-59A, *Computer-Aided Acquisition and Logistic Support*, 28 September 1990.
V. Berman, "An Analysis of the VITAL Initiative", *VHDL Boot Camp*, Proceedings of the VHDL International Users' Forum Fall Conference, 11-13 October 1993, San Jose, CA, VHDL International Users Forum, c/o Conference Management Services, Menlo Park, CA.
O. Levia and F. Abramson, "ASIC Sign-Off in VHDL", *VHDL Boot Camp*, Proceedings of the VHDL International Users' Forum Fall Conference, 11-13 October 1993, San Jose, CA, VHDL International Users' Forum, c/o Conference Management Services, Menlo Park, CA.

modeling is used to test how effectively test vectors detest logic-level faults.

Logic Value. For WAVES, A VHDL enumerated type that names all possible signal values that either can be applied to the external inputs of the MUT or can be sensed as external outputs of the MUT.

Match. The WAVES operation that samples the actual response of the MUT (or its model), compares it with the expected response, and produces a flag depending on whether the response was within the tolerances specified by the WAVES waveform generator procedure.

Microelectronic Devices. Monolithic integrated circuits, hybrid integrated circuits, radio frequency (RF) and microwave (hybrid/integrated) circuits, multichip microcircuits, and microcircuit modules. [MIL-HDBK-454]

Model Reference Library. An implementation-dependent storage facility for a set of executable models that can be simulated in a VHDL simulation environment. Models contained in a model reference library may not be provided in source form because of licensing and proprietary data restrictions. Such restricted models are to be avoided in DoD system designs because they may make the described equipment unsupportable in the long term.

Module. A synonym for a component.

Module Under Test (MUT). The component of a test bench that is being tested.

Object. A VHDL object contains a value of a given type. There are four classes of objects in VHDL '93: constants, signals, variables, and files.

Operating Condition. The physical and electronic environment in which physical components are designed to operate, such as temperature range, signal excursions, logic-level definitions, maximum power dissipation, and radiation hardness. An operating condition for a VHDL model is defined in terms of a set of parameters that specify the aspects of the environment and a specific set of values for those parameters.

Operating Point. A specific simulation condition selected from minimum, maximum, and nominal. [EIA] An operating point specifies the values for the operating condition parameters used in a simulation.

Package. A VHDL design unit that contains declarations and definitions. Packages are used to encapsulate definitions of data types, constants, type conversion functions, and utility functions so that these common definitions can be reused throughout a model or across several models.

Partitioning. The process of decomposing a component into its subcomponents.

Performance. A collection of measures of the quality of a design that relate to the timeliness with which the system reacts to stimuli. Measures associated with performance include utilization, throughput, and latency (response time).

Performance Model. A model with incomplete numerical and internal state precision used early in the design cycle to estimate utilization, throughput, and latency.

Period. The time from the beginning of a WAVES slice to the end of the slice.

Physical View. A view that specifies that relationship between the component model and the physical packaging of the component, such as relating port definitions in the component model to the signal and power pins in the physical implementation of the component.

Pin. An electrical connection to a physical component. Pins are classified as signal pins, power pins, or unconnected pins. [EIA]

Port. A VHDL signal that provides a channel for dynamic communication between a module and its environment. A port is a signal declared in the interface list of an entity declaration, in the header of a block statement, or in the interface list of a component declaration. In addition to the characteristics of signals, ports also have an associated mode that constrains the directions of data flow allowed through the port. [VHDL '93 LRM]

Port Interface List. A list of ports that declares the inputs and outputs of a block, component, or design entity. It consists entirely of interface signal declarations.

Power Pin. An electrical connection through which electrical power is supplied for the operation of a physical device. Power pin specification is necessary for the procurement of physical components, but it is not necessary for simulation in VHDL models. [EIA]

Prime Item. A configuration item is a technically complex item such as an aircraft, missile, launcher equipment, fire control equipment, radar set, or training equipment. A prime item requires a B1-level specification for development. The criteria used to consider a configuration item a prime item are described in MIL-STD-961.

Primitive Data Types. A data type that is one of the data types predefined by VHDL. The VHDL primitive data types are: INTEGER, REAL, TIME, CHARACTER, BIT, BOOLEAN, and SEVERITY_LEVEL.

Primitive Module. A leaf-level module in a design hierarchy.

Printed Circuit Boards. Boards used to mount components. A conductor pattern in, or attached to, the surface of the printed circuit board provides point-to-point electrical connections for the components mounted on the board. [IEEE]

Process. The basic mechanism in VHDL used to describe behavior. All concurrent signal assignment statements can be represented as equivalent processes.

Processor. A hardware component that has the ability to follow a program or list of instructions stored in a memory (RAM or ROM), which allows it to perform some detailed set of tasks. At the simplest level the instructions

may be just a set of parameters. At the most complex level, the instructions may be a compiled Ada program.

Processor-Memory-Switch-Level Model. A model that de-scribes a system in terms of processors, memories, and their interconnections, e.g., busses or networks.

Programmable Device. A hardware component whose be-havior can be altered after manufacturing of the device. Programmable devices include processors (whose be-havior can be changed by changing the program in memory), programmable logic arrays (PLAs), and field programmable gate arrays (FPGAs).

Prototype. An initial or early version of a system in a non-deployable form and usually created to validate certain aspects of the design. It may not have all of the function-ality, appearance, or internal complexity of the expected final design. For example, a computer simulation of a hardware component may be considered a prototype in which all physical and timing characteristics are not represented, but the functional characteristics are represented.

Race Condition. Occurs when the behavior of a device de-pends on the relative arrival order of signal values at a particular component of the device. Such differences may occur because two or more values are ultimately derived from the same signal by computations having potentially different delays.

Register Signal. A guarded VHDL signal that retains its last driven value whenever all of its drivers are disconnected.

Register-Transfer-Level Model. A model that describes a system in terms of registers, combinational circuitry, low-level busses, and control circuits.

Relevance. The component of a WAVES event value that is used to indicate the significance of the event to the sim-ulation. The possible values for a WAVES event rele-vance are predicted, observed, and required.

Resolution Function. A user-defined function used to com-pute the value of a signal that has multiple drivers. A resolution function is required whenever a signal has multiple drivers. The resolution function determines the value of the resolved signal as a function of the collection of inputs from the signal's multiple sources. [VHDL '93 LRM] It is invoked whenever the value of any drivers of the signals changes.

Scope. The range of VHDL text to which a declaration applies. For example, the scope of a declaration of an internal variable in a process include only that process.

Schematic Capture. The process of electronically drawing and storing a schematic diagram. The schematic cap-ture database can be used with simulation to verify design accuracy. [VLSI]

Sequential Logic. A logic relation in which the combi-nation of outputs of the relation is determined not just by the combination of current values but also by the his-tory of previous inputs to the relation.

Sequential Statement. A VHDL statement that occurs in the body of a process and is executed in the order in which it appears in the program and as controlled by the control statements of the process. Sequential state-ments are not executed concurrently.

Signal. A VHDL object with a present value, a past history of values, and a possible set of future values. Signals are objects declared by signal declarations or port declarations and are mechanisms used in VHDL to connect entities. VHDL processes or signal assignment statements create the possible future values of signals. Those connections to a signal that edit the future value of a signal are called drivers. A signal may have multiple drivers, each with a current value and projected future values.

Signal Pin. An electrical connection through which a com-ponent exchanges information with other components of the system. Specification of signal pins is necessary for both procurement and simulation. [EIA]

Signal State. The state of a WAVES event value determined by the logic level of the associated signal. A WAVES event can specify one of three logic levels: low, midband, and high. The midband value is used to indicate uncertainty about the value of the signal at the given time. A VHDL signal does not distinguish between state and strength, but data types can be defined for signals that do make this distinction.

Signal State/Strength Value. In VHDL, the encoding of the signal state and strength into a single value. The possible state/strength values for a VHDL signal can be described by an enumerated data type for the signal. IEEE Standard 1164 defines a standard enumerated type in its std_logic_1164 package. In WAVES, signal state and strength are treated separately.

Signal Strength. The ability of the specific WAVES signal driver to force a logic level in the face of conflicting logic levels from other signal sources. A WAVES event can specify signal strength as disconnected, capacitive, resistive, drive, or supply.

Simulation. The process of applying stimuli to a model over simulated time and producing the corresponding responses from the model at the simulated times at which those responses would occur in an effort to predict how the modeled system will behave.

Simulation Condition. A description of characteristics of the model used for a specific simulation. For example, a simulation condition would specify whether minimum, maximum, or nominal timing was to be used for the simulation and whether assertions on ports would be executed.

Simulation Model. A model that behaves or operates like a given system when provided a set of controlled inputs. [IEEE] A VHDL model that has been prepared for sim-

SUBJECT TERM (KEY WORD) LISTING

ASIC
Computer
Computer aided design
Design
Hardware simulation
Integrated circuits
Microelectronics
Modeling
Models
Simulation
Very high speed integrated circuits
VHSIC
VHSIC hardware description language
VLSI
Waveform and vector exchange specification
WAVES

Custodians:
Army - CR
Navy - EC
Air Force - 11
DLA - DH

Review activity:
Air Force – 19

Preparing activity:
DLA – CC

(Project 5962-1236)

SUPERSEDES PAGE ST-1 OF MIL-HDBK-62