

Teaching VHDL At The Undergraduate And Graduate Levels

J.R. Armstrong
F.G. Gray
Department of Electrical Engineering
Virginia Tech
Blacksburg, VA 24061

Abstract

Approaches to teaching VHDL at the undergraduate and graduate levels are given. An undergraduate course gives an introduction to VHDL and teaches high level design synthesis in a PC based schematic capture environment. A graduate course teaches the detailed syntax and semantics of the VHDL language and fully explores its modeling capabilities. Laboratory support and experiments are described for each course. Conclusions based on teaching experience are drawn.

Introduction

In this paper we describe approaches to teaching VHDL at the undergraduate and graduate levels. The approaches are derived from three years of teaching Computer and Electrical Engineering undergraduates and 6 years of teaching Electrical Engineering graduate students. Material taught in these courses have been transformed into two books. The first book, "Chip Level Modeling With VHDL", is a 148 page monograph devoted to VHDL behavioral modeling and was published in 1989 [1]. The second book, "Structured Logic Design With VHDL"[2], is a 482 page textbook which teaches VHDL in a design synthesis context.

Undergraduate Instruction

At Virginia Tech we teach a semester length undergraduate course (EE4506 - Digital Systems Design II) which covers Chapters 1,2,3,4,5, and 8 from "Structured Logic Design With VHDL". If time permits, material from Chapter 6 can also be added. The purpose of the course is teach how to do digital design at a high level of abstraction. Two main steps in this process are emphasized: 1)Development of a hardware description language model and 2)Transformation of the model into a logic circuit that can be fabricated. In teaching step one we use VHDL, the VHSIC Hardware Description Language because, in the opinion of the authors, it has the most comprehensive set of modeling constructs available in any hardware description language. In teaching the VHDL language in this course, one merely tries to give an introduction without studying advance features. A conservative, easily synthesizable algorithmic modeling style is emphasized. Step two of the high level design process involves synthesis. In this course, synthesis is viewed as a multi-step process, beginning with an English description which is transformed first into VHDL and then from VHDL into a circuit schematic. Wherever synthesis

is discussed, emphasis is placed on understanding the relationship between VHDL language constructs and the implied logic circuit. The course is the second course in a two semester logic design sequence. Thus the students have had a course in traditional gate level design techniques when they enter the course described here. The students are either juniors in Computer Engineering, for whom the course is required, or Electrical Engineering seniors, for whom the course is an elective.

To support the course in the laboratory, we use a PC version of ViewLogic Inc.'s Workview. While not implementing the full VHDL language, PC Workview provides for the preparation of algorithmic VHDL models, using a subset of VHDL which we have found to be sufficient for beginning students (Later versions of PC Workview may offer full VHDL, but at the time of writing this paper, this was not the case). Workview also provides schematic capture and waveform display capability; schematics and waveforms can be printed on standard PC printers. Students work six assignments in the Workview environment:

1. An introductory assignment to familiarize themselves with Workview's schematic capture environment.
2. An assignment to develop and simulate a single VHDL behavioral model.
3. An assignment to develop a model of a counter, or some similar circuit. Students develop VHDL behavioral models for counter flip-flops and gates, and use the schematic capture capability of Workview to construct a structural model for the counter.
4. Three assignments which require translation of English descriptions into VHDL behavioral models which are then simulated. The VHDL behavioral model is then translated to a structural equivalent using gates and flip-flops. In this case, the built-in Workview primitives are used. The structural model is then simulated to verify that its response matches that of the behavioral model. Students are graded on their imaginative use of the Workview system in exercising the model and displaying their results. These assignments reinforce the principles of synthesis taught in the lecture.

All students in our department have their own PC, so the use of a PC based system such as Workview is effective in being able to serve the 80 students we normally teach in our second digital design course. To acquaint students with some of the language features not covered in the full language subset, we also give them at least one assignment on a VHDL software system that implements the whole language.

Graduate Instruction

The VHDL language is the main focus of a graduate course, Digital System Modeling With Hardware Description Languages (EE 5514). In this graduate course, the first eight chapters of "Structured Logic Design With VHDL" are covered in one semester. In language terms, the emphasis is on covering the broad range of constructs in the language and examining in detail the language semantics. Thus it is important to use a VHDL software system which implements the whole language. We use the Synopsys Sun based VHDL system and Model Technology's PC VHDL software for this purpose.

For this course the student's laboratory assignments include:

1. An assignment to develop and simulate a single VHDL behavioral model.
2. An assignment to develop a model of a counter or some similar circuit. VHDL behavioral models are developed for counter flip-flops and gates, and then a VHDL structural model

- is developed for the whole system.
3. An assignment involving complex data types, e.g., using array aggregates and record types to implement a tabular representation of a finite state machine.
 4. A system modeling assignment that involves the use of bus resolution and bus protocols. The URISC system discussed in Chapter 6 of "Structure Logic Design With VHDL" is a good example of this.
 5. A semester project where the students model a system of their choice. One can choose projects which stretch the language, i.e., involve applications which are not typical, such as modeling of parallel processing systems or systems which are not digital.

Other Books:

Most of the material for our courses comes from "Structured Logic With VHDL" and "Chip Level Modeling With VHDL" because they evolved from our course notes. At this time there are many other good books dealing with VHDL [2,3,4,5,6,7,8,9,10] that contain useful material for classroom use. Different points of view are important. We always have list of such books on reserve in the university library and draw on them for lecture examples and exam questions.

Conclusions

Our experience in teaching VHDL has taught us the following:

- 1) Both undergraduate and graduate students adapt quite easily to VHDL. However, the prerequisite background is important: programming in a high level programming language and courses in computer organization and logic design.
- 2) While undergraduates seem to enjoy the language, they get bored with prolonged exposure to VHDL abstractions. Graduate students, because of their greater maturity have less trouble with this.
- 3) In a graduate VHDL course, getting an A grade can be more difficult than in some other courses the students take. The subset of students who seem to have real mastery of the VHDL course material, and thus deserve an A is smaller than in some other graduate courses we have taught. One contributing factor to this situation is that students whose understanding of English is limited can have difficulty understanding VHDL semantics.
- 4) Debugging simulation models and using schematic capture systems is hard work! When making up homework assignments, it is important to do them yourself so that you know that the length of time to complete the assignment is reasonable.
- 5) There presently is some overlap between our graduate and undergraduate courses. This will disappear as more entering graduate students have undergraduate VHDL background. The graduate course will become more of a comparative course where other languages such as Verilog and UDL/I are also treated.
- 6) Most universities get their CAD software by getting donations or paying the university price. In order to get a donation, a proposal may have to be written followed by several months of negotiations. A considerable amount of faculty time is spent on this. In fact, new faculty may not even do it because it takes time from the activities that are higher valued when it comes to promotion and tenure. Companies spend time administrating their

university programs, and some are very concerned that the program enhance their competitive advantage. In fact, an overly structured program can hurt the company if it's too much work to get the software. University faculty are going to select software based on its features, not on the nature of the university program. Also most universities will be using a number of VHDL products to meet their needs. It's not reasonable to expect them to restrict themselves to just using one. The best situation would be a vendor pool of VHDL software where any professor could select what he or she wants. It would save everyone a lot of unnecessary work. Most companies charge annual fees of approximately \$2000. These are reasonable for university budgets as long as the number of workstations or PCs allowed is not too small. Digital design classes in universities can be large.

- 7) From an educational point of view, VHDL is an important development. It allows us to teach both top down and bottom up logic design in a structured way. The nature of the language also teaches students many important features of programming languages, data types and data structures. This is required knowledge for the workstation environment that they will be working in when they leave the university.

References:

- 1) *Chip Level Modeling With VHDL*, J.R. Armstrong, Prentice Hall, 1989.
- 2) *Structured Logic Design With VHDL*, J.R. Armstrong and F. G. Gray, Prentice Hall, 1993.
- 3) *VHDL: Analysis and Modeling of Digital Systems*, Z. Navabi, McGraw Hill, 1993.
- 4) *VHDL*, D. L. Perry, McGraw Hill, 1991.
- 5) *VHDL: Hardware Description and Design*, R. Lipsett, C. Schaefer, and C. Ussery, Kluwer, 1989.
- 6) *The VHDL Handbook*, D.R. Coelho, Kluwer, 1989.
- 7) *VHDL Designer's Reference*, J. Berge, A. Fonkoua, S. Maginot, and J. Roulliard, Kluwer, 1992
- 8) *A Guide to VHDL*, S. Mazor and P. Langstraat, Kluwer, 1992.
- 9) *Applications of VHDL to Circuit Design*, R. Harr and A. Stanculescu, Kluwer 1991.
- 10) *Performance and Fault Modeling With VHDL*, J. Schoen, Prentice Hall, 1991.