

# Experience in Training 100 VHDL Users in 100 Days

Jackie Patterson

Amdahl Corporation

1250 East Arques Avenue

Sunnyvale, CA 94088

## Abstract

After two small pilot projects using the VHDL language in a top-down design methodology, Amdahl began its first major project in VHDL. Since the project was staffed with experienced engineers with limited VHDL exposure, and VHDL is a complex language, training was on the critical path. Amdahl embarked on the systematic training of a large and diverse development team in a period of several weeks. This paper will describe the process from start to finish, covering the need for training, course content, instructor selection, and post-training support. Key successes and lessons learned will be highlighted as well.

## 1 Introduction

Amdahl's experiments with VHDL began with the design of a small piece of a computer. This project involved a handful of hardware designers writing and simulating a high-level VHDL model of the design, and then evolving it manually into gates. The second project undertaken using VHDL was the design of a major subsystem of an Amdahl mainframe. Roughly one dozen hardware designers wrote an RTL-level model which was simulated and synthesized into gates. VHDL began spreading throughout the ranks of the Amdahl engineers as other smaller projects adopted it. Training courses were set up on an as-needed basis.

Ultimately, a decision was made to do a whole computer design in VHDL. Since this would involve tens of hardware designers, tools support people, and dedicated design verification engineers, most of which had not been exposed to VHDL, a VHDL training program was arranged.

This paper will cover the issues dealt with while setting up the training program, and present some preliminary results of the training.

## 2 Need for Training

VHDL is a very complex language for specifying hardware designs. It features strong type checking, generics, data hiding, and other constructs which are more typical of a high level programming language than a designer's toolkit. These more advanced constructs are not necessarily obvious to engineers familiar mostly with hardware, and with limited experience in C or Pascal. A hands-on training course is useful in bringing home these new ideas.

Granted there are those engineers who prefer to learn VHDL by reading a text. In many cases they elected to attend the formal training class just to be sure they got everything. One technical lead cited the need to know what others were covering in the class in order to be more effective when coaching junior engineers.

### **3. Course Descriptions**

The appropriate course content was determined by experiences with previous VHDL projects. Initially, engineers were sent to a VHDL Synthesis course which focused on the use of the tools, and the synthesis policy. It covered the VHDL language itself in an incidental fashion. It appeared that this was not enough time spent on the language, so the same group of engineers attended a dedicated language course. This turned out to be too slow paced for engineers now familiar with the synthesis tools. We also sent a group of engineers to a dedicated language course only, with limited coverage of simulation and synthesis tools. This left the engineers to learn the tools on their own, which resulted in a wide variation in knowledge between the engineers. Hence we settled on the following sequence:

1. Send engineers to a week-long dedicated VHDL language course, with a light treatment of synthesis tools added at the end.
2. Allow them time to experiment with the language and the synthesis tools.
3. Follow up with a Question and Answer session on the synthesis tools.
4. In parallel, managers and others needing an overview of VHDL attended a one-day high-level lecture on VHDL.

The actual topics covered in a dedicated VHDL language course did not vary much from course to course, so they will not be outlined here. One requirement worth noting is the ability to do hands on work with VHDL during the course, writing code, compiling, simulating, and synthesizing it. This aids in developing a concrete understanding of the language.

Of course, feedback from the students was incorporated into later versions of the course. Also, any Amdahl-specific information brought up in class, such as locations of libraries, who to call for help, etc., was added to the course slides. In this way, the course continues to evolve over time to suit Amdahl's needs.

The typical VHDL class size was limited to 25. This size allowed us to move students through reasonably quickly while still providing individual attention. The VHDL Overview course had over 100 students.

Initially, we had much discussion over the best way to group students: either in homogeneous groups or a cross-section of different work areas. Homogeneous groups are easier for the instructor to deal with, however, the cross-section allows people to learn a little about areas other than their own. In the end, the politics of scheduling class space forced us to form classes that were of the cross-section variety. This did not cause problems in the actual classes.

### **4. Choosing the Course Provider**

The following factors influenced our choice of instructors for the class:

- Instructor quality.
- Minimize overall cost.

- Flexibility in scheduling courses at the provider's site or Amdahl's site.
- Flexibility in tailoring the class to Amdahl's specific needs of three days on the VHDL language, and two days on synthesis.
- Course taught on Synopsys tools to minimize confusion when engineers returned from class to use Synopsys synthesis and simulation tools.
- Capability to teach several classes within the next several weeks.

After reviewing quotes from several sources -- tool vendors, universities, consultants -- Synopsys was chosen to give the classes. Their standard VHDL System Design and Simulation Workshop focuses on the VHDL language and formed the basis of the course for Amdahl. Material from the standard VHDL Synthesis Workshop was added to round out the course. The one-day overview, the standard Synopsys course titled Making the Transition to High-Level Design, was also given at Amdahl.

## 5. Post-training Support

There were three aspects to Amdahl's post training support. The first was an extended homework assignment. Each hardware designer was asked to code a section of logic they were familiar with in VHDL. Additional simulation and synthesis seats were provided for them to experiment with after the course. This was very effective in getting people immersed in the language. Then once they had the language down, they could focus on the real design effort without being distracted by looking up language constructs.

Secondly, Amdahl has an internal VHDL User's Group which meets weekly to share information between those working in VHDL. Training course attendees were encouraged to attend the user's group meetings to benefit from other's experiences. It turned out that attendance in the meeting was initially overwhelming, but quickly dropped off to a small core of people interested in working issues. As such, it may not have been as effective in spreading VHDL knowledge as we hoped. However, informal networks sprung up to fill the gap.

In addition to the manuals given out during the course, a VHDL text was provided. Unfortunately, we were unable to find a good VHDL reference manual. The LRM is too obtuse, while most texts are just that, a text book, and not as useful as a reference unless one has read the whole thing. Nonetheless, VHDL: Hardware Description and Design, by Lipsett, Schaefer, and Ussery was selected, and given to each student.

## 6. Summary

By following the above mentioned steps, Amdahl was able to train 100+ engineers within a three month period, and have them productively working in VHDL.

## 7. Acknowledgments

I would like to thank Bob Knight, Angel Rampy, and the whole training team for pulling this training effort together.