# Using HDLs in an Introduction to Digital Electronics Course

Dr. Richard Spillman
Pacific Lutheran University

ABSTRACT

The gap between digital electronics as taught in the academic world and digital electronics as it is practiced in industry has widened in the last few years. One aspect of that gap is the use of HDLs in industry. While HDLs are growing in their use in industry, few courses in digital electronics systematically teach students how to design with HDLs. This paper reports on the use of both Verilog and VHDL as a central feature of the digital electronics course at Pacific Lutheran University. Both languages have been taught at different times over the last few years. Initially, Verilog was selected over VHDL because of its close connection to C, which many of the students have used in other classes and because of the close connection between schematics and low level Verilog code. Now, the course is moving towards increasing use of VHDL. The paper outlines the need for HDLs in the introductory digital electronics course and explores the advantages and disadvantages of both Verilog and VHDL. The paper also reports on the the approach used to introduce HDLs to the students and the development of projects requiring HDLs. The results both in terms of student success in the class and reports from students who have graduated are summarized.

## Introduction

Hardware Description Languages, HDLs, have emerged as a common tool to aid in the design of digital circuits. Initially, they were used for both simulation and documentation. Now they are also used for synthesis as well. There are a variety of HDLs but two now compete as industrial standards: Verilog and VHDL. While industry makes use of these HDLs, it is still the case that many engineering students are not exposed to either language at all and others only see them during the last stages of their education. At Pacific Lutheran University, both Verilog and VHDL have been used in the first course on digital electronics. Upper division courses assume an initial exposure to an HDL which allows these courses to explore simulation and synthesis in greater detail. This short paper discusses the advantages and disadvantages of the two HDLs and reports on their use in the beginning digital electronics course at PLU.

## Using an HDL

Both Verilog and VHDL have been used in the first digitial electronics course, Engr 346, taught at PLU. This course is required by all electrical and computer engineering students and all computer science students. It is usually taken in the second or third year of a students program, though sometimes advanced freshman elect to take the class early. Some computer science students put it off until their senior year. As a result, the class consists of a mix of majors at different stages in their program. For most students it is their first true design course.

Over the last eight years, an HDL has been part of the course content. For the first 6 years the HDL was Verilog and for the last two years it has been VHDL. Regardless of the specific language, the nature of the assignments remained the same. HDLs were introduced by the fourth lecture of the semester long course, just after the basic steps for combinational design were covered. At this point students knew how to design a simple circuit using logic gates so the HDL was introduced as a method of describing the circuit which allows for easy simulation. In this way, the structural modes of either Verilog or VHDL become part of the learning process. From the very beginning of their education students design circuits using both schematics

and HDL structural code together. Every design assignment, both in class and in the lab, requires that both be done.

**Comparison of HDLs**

While the users of Verilog and VHDL are engaged in a (often friendly) competition to determine which is the best HDL for circuit design and simulation, the truth of the matter seems to be that both languages are capable of implementing any circuit. True, some circuits may appear to be "easier" to implement using Verilog while other circuits may be "easier" to implement in VHDL. Overall, the choice of language is made most often by the availability of previous code and the training of the designers. If a given project has prior or related code done in Verilog and the designers have experience in Verilog (and Verilog tools are available), then the design is completed in Verilog. The availability of tools which allow for mixed Verilog and VHDL designs may change this somewhat, but the issue of language familiarity will remain as perhaps the major determinate of language use.

In an educational setting where the students come into a first level design course without any experience in an HDL, the issue of which language to use is more open. The criteria at PLU for language choice was two-fold. First, tool availability was the major issue. Students must have easy access to a new language – access that allows them to explore language concepts and experiment with language implementation. In today's, environment in which many students have their own PC, this means that they should have affordable access to a personal copy. Second, ease of learning is criteria for language selection. Given only one semester to learn a language in a course in which the language itself is secondary to the main instruction goals, the students must be able to quickly learn the basics of the language and experience success in applying it to the task of circuit design.

Based on these criteria, Verilog was selected as the initial language. Early on (1991 to 1994) Verilog was available through the educational program at Cadence. Though this software did not run on PCs, fewer students had their own PC during those years. So, personal copies were less of an issue. From 1994 to 1996, when PC ownership among students at PLU increased, a free student version of Verilog was available from VeriWell. In addition, the instructors in the program felt that since Verilog was closely related to C, a language which most of the students at PLU were taught, that the learning curve for Verilog would be shorter than VHDL.

In the last two years we have switched to VHDL. Using the Altera MAX+PLUS II software, students have access to a free version of VHDL. The MAX+PLUS II system is intended for CPLD design but the schematic capture and VHDL compiler included in the package make it an excellent tool for teaching VHDL development. The software is free to students and runs on a PC. In addition, it has been our experience that VHDL is as easy to learn as Verilog. In other words, the advantage of prior knowledge of C does not seem to significantly improve a students ability to learn Verilog. Hence, for the last two years, VHDL has been the language of choice for the introduction to digital electronics course.

**Using VHDL**

The lectures on VHDL (in fact, the entire set of digital electronics lectures) are available in power point format from the author,s web site (www.plu.edu\~spillmrj). VHDL is introduced during the fifth lecture (each class is 1.5 hours long so this represents the beginning of the 3rd week of a typical four credit semester class). By this time the students have a basic understanding of the fundamentals of combinational logic design. The initial lecture is designed to introduce the structural options available in VHDL. For example, Figure 1 is the slide used to illustrate the two components of all VHDL code: the entity and the architecture segments.

The entity segment is easy to visualize since it describes the inputs and the outputs of the circuit. For students that have programming experience (which is assumed of all students in this class), the entity segments is much like the variable declaration statements of PASCAL or C. The architectural segment is more complicated because it can be implemented using several different approaches: data flow, behavorial,

structural, or some combination of the three. Given that the students are familiar with the structure of a combinational circuit, the initial look at a VHDL program is given in the form of a structural architecture as illustrated in Figure 2. MAX+PLUS II is the tool used to enter, compile, and simulate a VHDL file. Students are shown how to enter their code in a MAX+PLUS II text file, take advantage of the VHDL templates provided by the package, and then compile the code (see Figure 3).
Once the code has been compiled, they are shown how to generate a symbol for their circuit and how to use the waveform component of MAX+PLUS II to simulate the operation of the circuit (see Figure 4).

**Results**

After the first two lectures on VHDL, students are required to create VHDL simulations of all lab projects and most of the designs which are assigned as exercises. VHDL becomes an integrated part of all remaining lectures. When a new design feature is introduced, the use of VHDL code to implement that feature is also covered. For example, when MSI devices such at MUXs and decoders are covered, the basic VHDL code for these devices is also presented. As the students become comfortable with writing simple structural models, the power data flow and behaviorial architectures is introduced. By the end of the class, students see design as more than just producing a circuit schematic – an HDL has become an important tool for design.

As a side note, the use of MAX+PLUS II is useful because when the class looks at PLDs, they discover that they have been using a PLD design tool all along so the transition to PLD, CPLD, and FPGA designs is easy to make.

RICHARD SPILLMAN
Dept of Engineering
Pacific Lutheran University
Tacoma, WA 98447
Spillmrj@plu.edu

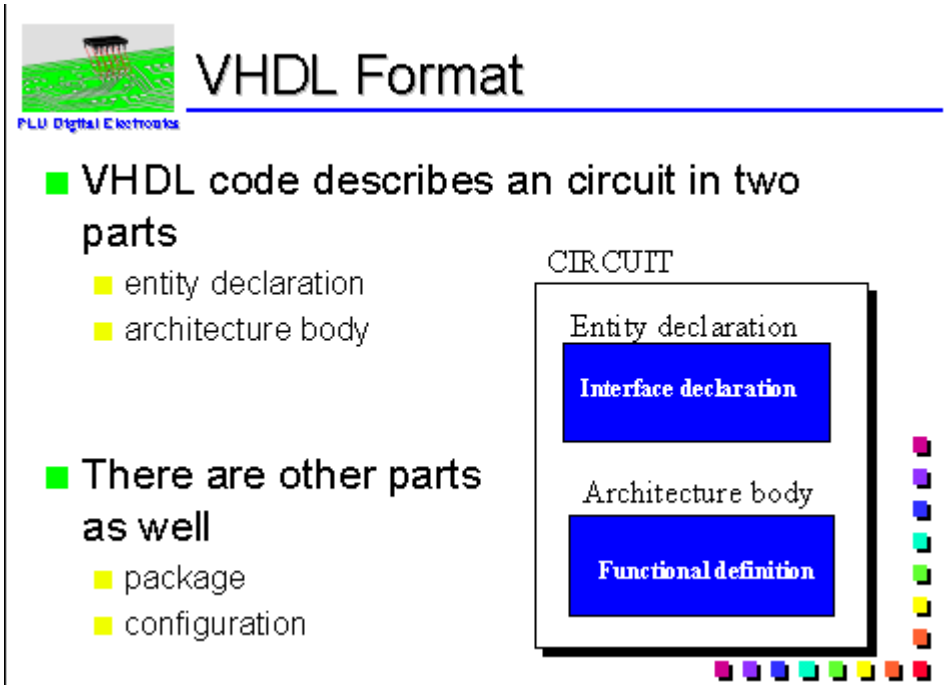



**FIGURE 1: VHDL Components**

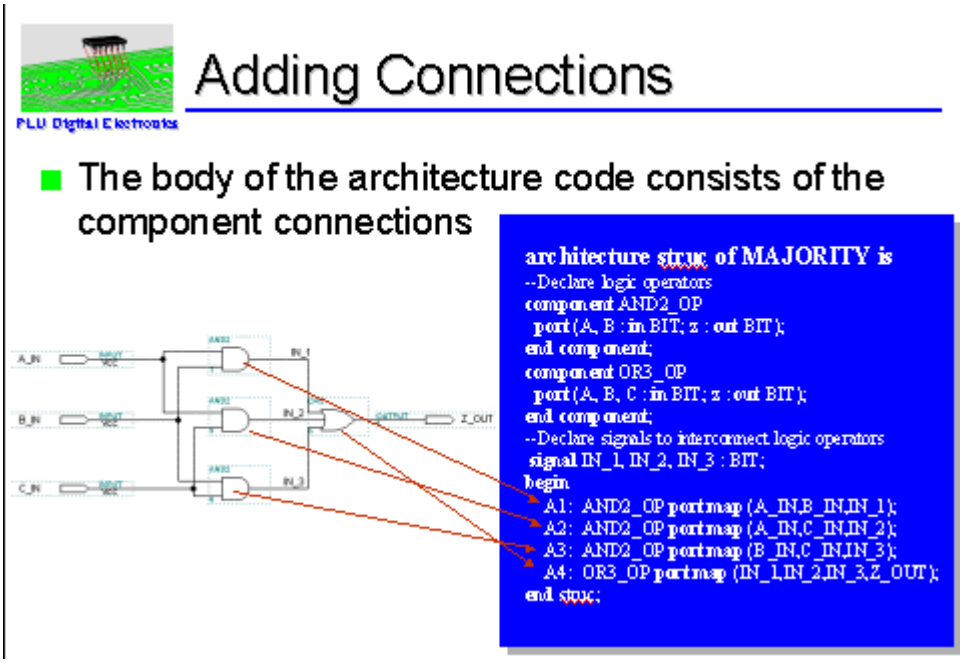
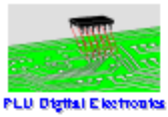



**FIGURE 2: Structural Architecture**


**FIGURE 3: MAX+PLUS II Templates**




# Simulation Results

- A successful simulation run:


The new waveform window:

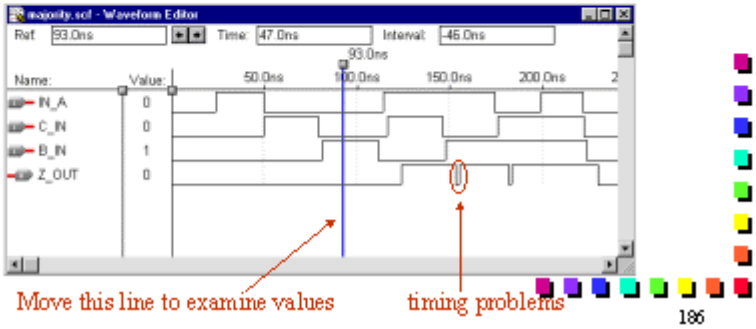





**Figure 4: Simulation in MAX+PLUS II**