Course: Digital System Design with VHDL

Instructor: David M. Clark

Prerequisites: High-level programming language (e.g., C, C++, Pascal)

Equipment: In order to run the required design tools (Cypress Warp2 VHDL Compiler for PLDs), students will be required to have their own access to a Personal Computer with the following system requirements:

- IBM PC or equivalent (486 or higher recommended)
- 16 MBytes of RAM (32 MBytes recommended)
- 100-MBytes hard disk space
- CD-ROM drive
- Two- or three-button mouse
- Windows 3.1x, Windows 95, or Windows NT
- 3.5" floppy disk drive

Introduction

The *VHSIC*[*] *Hardware Description Language* (VHDL) is an IEEE standard language for describing hardware in terms of high-level, software-like constructs. VHDL is a highly structured, strongly-typed language that is well suited to a multi-user, shared development environment. In addition, the language is designed to accurately represent the concurrent nature of hardware though running on sequential computers. ([*]*VHSIC* stand for *Very High Speed Integrated Circuits*.)

VHDL enjoys widespread support within industry, with commercially available compilers, simulators, and synthesis tools. Virtually all engineering companies throughout the world utilize hardware description languages such as VHDL to perform digital design.

Similar in many ways to object-oriented programming languages, VHDL contains high-level coding features to facilitate design reusability, maintainability, readability, and concurrent engineering. Design cycle-time reduction is a primary benefit of VHDL, especially when coupled with an organized effort toward design reuse.

VHDL allows designs to be represented at any conceptual level, be it a simple programmable device (PLD, FPGA), an integrated circuit (ASIC), a circuit card (board), or even a multi-board system with off-the-shelf parts. In addition, each and every level can be simulated and custom designs can be synthesized directly into gates within a target technology.

Course Highlights

This course teaches the basic VHDL language syntax and the use of VHDL for designing digital hardware.

The focus of this course is the synthesizable subset of VHDL, but various non-synthesizable constructs are discussed in the context of testbench generation for debugging and design verification.

Hands-on design, coding, and simulation with VHDL are integral parts of this course. Synthesis of VHDL will also be discussed and examples presented.

The **Cadence Leapfrog** VHDL simulator will be used extensively for both homework and a final project.

Coding styles and methodologies are also introduced.