SUTHERLAND HDL

22805 SW 92nd Place., Tualatin, Oregon 97062 • Phone: +1-503-692-0898 • URL: www.sutherland-hdl.com

Advanced Verilog-2005 PLI 2.0 (VPI) and SystemVerilog DPI

Overview

The Verilog Programming Language Interface (PLI) is an important part of Verilog design. The PLI provides designers a means to extend the Verilog language, and to customize Verilog software tools to perform specific verification tasks. A basic premise of the creators of Verilog was to keep the Verilog language directly related to hardware design, and to provide a procedural interface to tie verification and abstract modeling tasks into a Verilog simulation.

Advanced Verilog-2005 PLI 2.0 (VPI) and SystemVerilog DPI is a comprehensive workshop on the IEEE 1364-2005 Verilog Programming Language Interface. The workshop teaches how to write PLI applications to extend the capabilities of Verilog software tools by reading test vector files, verifying test coverage, calculating power usage, annotating accurate delays, and interfacing to Bus Functional C models. Students also learn how to use the IEEE 1800-2005 Direct Programming Interface (DPI) that extends the Verilog PLI. About fifty percent of the class time is devoted to hands on experience writing several useful PLI applications.

Students receive a *comprehensive student guide*, and a handy *"Verilog PLI Quick Reference Guide"* ($30 value).

Course Objectives

Students will learn how to write PLI DPI applications that are portable across all major simulators. Students will also understand the appropriate times to use the Verilog PLI and when to use the SystemVerilog DPI.

Intended Audience

This advanced workshop is for software engineers, verification engineers and hardware engineers who will be writing or maintaining PLI applications.

Prerequisites

Knowledge of the C programming language is mandatory — All labs involve writing small C programs. A basic knowledge of the Verilog HDL language is also mandatory.

Presented By

Stuart Sutherland. Mr. Sutherland is an expert engineering consultant, with extensive experience in Verilog design, simulation and verification. He is the author of *"The Verilog PLI Handbook"*, a comprehensive reference book on the PLI, and has served as the chairman and technical editor for the IEEE 1364 Verilog PLI standard.

Software Tools Used

During class labs, students will write PLI and DPI applications in C, and link these applications to Verilog simulators. The course includes information on using the Cadence *NC-Verilog*™, Synopsys *VCS*™, and Mentor Graphics *ModelSim*™ simulators.

Workshop Locations

This workshop can be presented on-site, at your facilities. We also offer several open-enrollment workshops throughout the year. For more information, please refer to our web page, *www.sutherland-hdl.com*, or call us at +1-503-692-0898.

SUTHERLAND HDL

22805 SW 92nd Place., Tualatin, Oregon 97062 • Phone: +1-503-692-0898 • URL: www.sutherland-hdl.com

Syllabus — Advanced Verilog-2005 PLI 2.0 (VPI) and SystemVerilog DPI

Day 1

Introduction to the PLI standard
- The purpose of the PLI
- The IEEE PLI 1.0 and 2.0 standards

VPI task function callbacks
- User-defined system tasks and functions
- Calltf and compiletf applications
- The VPI interface mechanism and callback registry
- Lab: compile and link a PLI application

Using VPI routines
- VPI object handles
- VPI object relationships
- VPI object properties
- Lab: write C programs to access task arguments

VPI routines to access object properties
- Reading integer and boolean properties
- Reading string properties
- Lab write a C program to display design data

Traversing a simulation data structure
- Object relationships
- Understanding the IEEE 1364 object diagrams
- Traversing single levels of design hierarchy
- Traversing multiple levels of design hierarchy
- Lab: write a C program to extract data on ASIC cells

Day 2

VPI routines to read values
- Reading logic values
- Reading delay values
- Lab: display design debugging information

VPI routines to modify values
- Modifying logic values
- Modifying delay values
- Lab: write system function to calculate square roots

VPI routines to schedule PLI callbacks
- Simulation event callbacks
- Simulation time callbacks
- User-defined work areas
- Lab: write a C program to read test vector files

Interfacing to C models using VPI routines
- Callbacks based on value changes
- Passing inputs and outputs to C models
- Lab: write an interface to a C model

The SystemVerilog Direct Programming Interface
- How the DPI works
- Importing C functions into Verilog models
- Exporting Verilog tasks and functions to C
- When to use the PLI and when to use the DPI
- Lab: Directly importing a C function into Verilog