Sutherland HDL

Transitioning To The New PLI Standard

(or would you rather fight than switch?)

Stuart Sutherland

Sutherland HDL Inc.

Portland, Oregon

International Verilog Conference, March 1998

Sutherland HDL

For The Full Paper:

- The full text of this paper is available on the web:

www.sutherland.com/IVC98_PLI_paper.htm

- Could not be published in the IVC/VIUF conference proceedings because of restrictive IEEE copyrights

Sutherland HDL

Who Am I?

- I'm Stuart Sutherland
  - A Verilog designer
  - A Verilog consultant
  - A Verilog software tool specialist
  - A member of the IEEE-1364 standards committee and editor of the IEEE PLI Language Reference Manual
- I developed and teach an advanced PLI training course
  - Includes the old PLI standard
  - Includes the new PLI standard
- I have seen several issues with transitioning to the new PLI standard

Sutherland HDL

Overview of This Presentation

- There are two generations of the PLI standard
  - The original Verilog-XL PLI *(non)*standard
  - The new IEEE-1364 PLI standard
- Transitioning to the new PLI standard may be difficult
  - It is easier to use and it is harder to use
  - Application developers will have a learning curve
  - Not every simulator supports the new PLI standard
- ***So, should you use the new PLI standard?***
  Or should you stick with the old, tried and true standard?

Sutherland HDL

Objectives

- This presentations will answer the following questions:
  - What are the two PLI standards?
  - What are the advantages of the old PLI standard?
  - What are the advantages of the new PLI standard?
  - Should you switch the new PLI standard?
  - When is the best time to switch?
  - What could happen if you don’t switch?

What is the PLI?

- The **Programming Language Interface (PLI)** is:
  - A procedural interface between Verilog simulations and other software programs
  - Verilog models and stimulus can invoke other software programs during simulation
  - Other software programs can read/modify Verilog values during simulation
    - Delay values
    - Logic values
    - Design structure (read only)
    - Statement execution

How the PLI Works

- The PLI provides a way for users to create new Verilog programming statements
  - Called ***"user-defined system tasks and functions"***
  - Must start with a dollar sign (**$**)
- When the task or function is executed during simulation:
  - A user's C program is executed
  - The C program can read and modify what is happening in the simulation

```
initial
  $my_waveform(clock, data, q);
```

```
always @(posedge clock)
  cosine = $cosine(angle);
```

The PLI Standard

- The Verilog PLI standard has two main parts:
  - A library of C-language functions
    - Used in C application programs
    - Provide a means for C applications to access the internal data of a Verilog simulation
  - An interface mechanism
    - Links a Verilog simulator and a PLI C application together
    - Controls how the Verilog simulation and the PLI application interact

Sutherland HDL

The PLI History

- **1985** - The PLI was introduced by Gateway Design Automation as part of the proprietary Verilog-XL simulator
- **1990** - The PLI standard was released to the public domain along with the Verilog HDL and Verilog SDF standards
  - Open Verilog International (OVI) “owned” the Verilog HDL
  - The OVI version of the PLI was called **"PLI 1.0"**
- **1993** - OVI released **"PLI 2.0"**
  - A completely new interface, *intended to replace PLI 1.0*
- **1993** - OVI submitted Verilog to the IEEE for standardization
- **1995** -The IEEE-1364 Verilog PLI standard was released
- **1998**(proposed) -The IEEE-1364 Verilog HDL & PLI standard will be updated, with *many new features*

The IEEE-1364 Verilog PLI Standard

- The IEEE-1364 Verilog PLI Standard combines the OVI PLI 1.0 and 2.0 standards into one standard
  - IEEE-1364 PLI **TF** and **ACC** routines:
    - Are based on the OVI PLI 1.0 standard
    - Provide backward compatibility with existing applications
  - IEEE-1364 PLI **VPI** routines:
    - Are based on the OVI PLI 2.0 standard
    - Provide future growth for new PLI applications

The VPI routines are what I am calling "the new PLI standard"

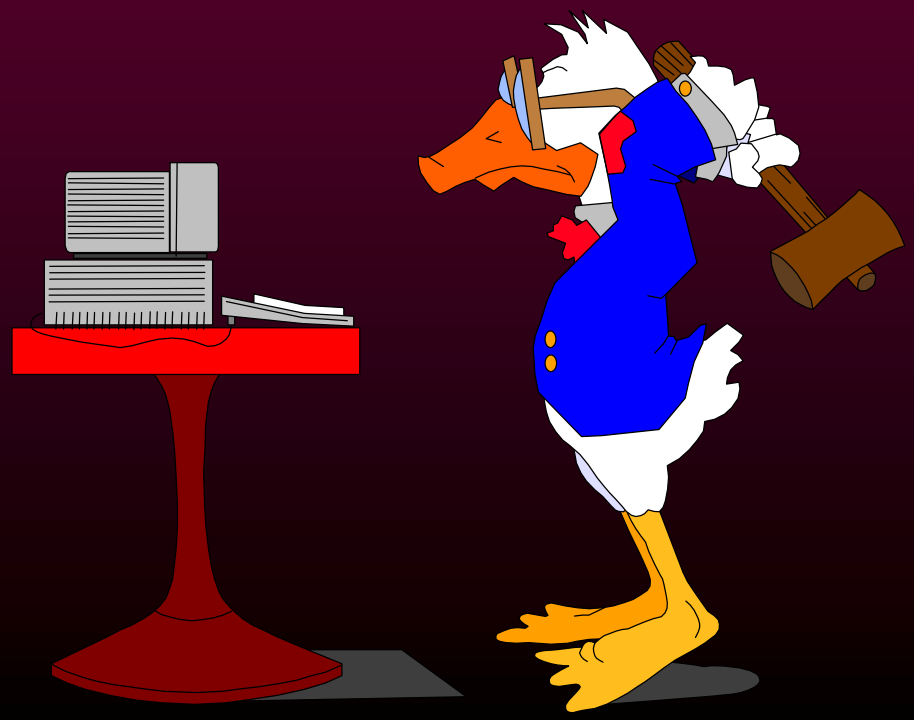
The "Old" PLI Standard

- The older PLI standard uses the IEEE TF and ACC routines
  - Has been in use for more than 13 years
  - Virtually every commercial and proprietary PLI application is written with the TF and ACC routines
- Advantages:
  - Supported by all major Verilog simulator products
  - Many experienced engineers know the standard
  - Extensive library of access routines
    - Easy to get to simulation data
    - Easy for ***hardware engineers*** to write "quick-N-dirty" applications

Disadvantages Of The "Old" PLI Standard

- The TF and ACC routines have several disadvantages:
  - Inconsistent syntax
    - Evolved over many years
    - Not developed to a specification
  - Many poorly documented "features"
    - Behavior of many IEEE routines not well defined
    - Several routines in Verilog-XL are not public domain
  - Many limitations in capabilities
    - Cannot access RTL modeling statements
    - Cannot control test bench and model execution flow
    - Limited (and inefficient) access to memory models

As a standard, PL I 1.0 was garbage!



Sutherland HDL

The "New" PLI Standard

- The new PLI standard adds the IEEE VPI routines
  - Documented in 1995 (3 years ago)
  - No commercial PLI application have been written with the VPI routines
- Advantages:
  - Consistent syntax, written to a well-thought specification
  - Thoroughly documented
  - No limitations -- can access everything in Verilog models
  - Forces good C coding style (structured programming)
  - Will access all new features coming in the proposed 1998 Verilog standard

New Language Features Only Supported by the VPI Standard

- Enhancements for modeling testing, design management, and Intellectual Property
  - Verilog configurations
  - Generate statements
  - Multi-dimensional arrays
  - File I/O enhancements “In a big way”
  - Signed arithmetic
  - Recursive tasks and functions

New Language Features Only Supported by the VPI Standard

- Enhancements for deep submicron design
  - On-detect glitch propagation
  - X-pulse glitch timing
  - Negative setup and hold times
  - Recovery and removal timing constraints
  - Timeskew and fullskew timing contraints

Disadvantages Of The "New" PLI Standard

- There are some disadvantages to using the VPI routines:
  - Difficult to learn
    - Very different from the old TF and ACC routines
    - Few engineers know the standard
    - Some find the documentation style non-intuitive
      - VPI uses data diagrams instead of text descriptions
    - Few examples to draw from
  - RISC-like library of routines make it harder to access data
    - Can require several VPI routines to do the same functionality as a single ACC routine
  - Not supported by most major Verilog simulator products

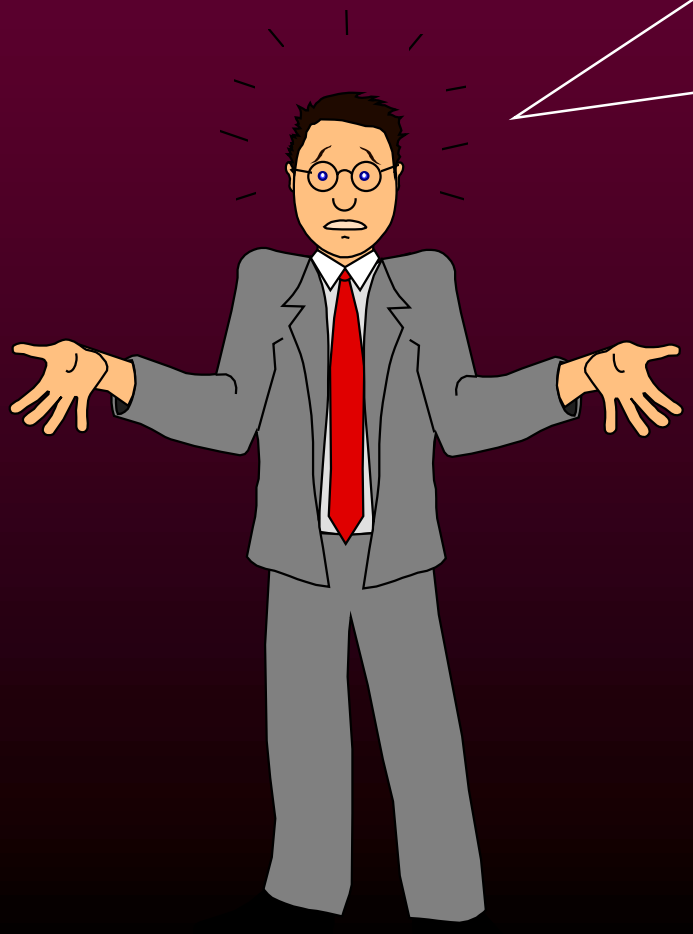
Who Supports VPI?

- Most major Verilog simulators do not currently the VPI routines
  - Cadence Verilog-XL, 100%
  - Cadence NC-Verilog, 100%
  - Viewlogic VCS, vague plans* (sometime in 1999?)
  - Mentor/Model Tech ModelSim (was Vsystem), no plans*
  - Avanti Polaris (was Purespeed), no plans*
  - VeriBest VeriBest / Fintronic FinSim, no plans*
  - Simucad Silos III, no plans*

*disclaimer: based on author's opinion, may not reflect official marketing positions

So Why Use The New PLI?

Sutherland HDL



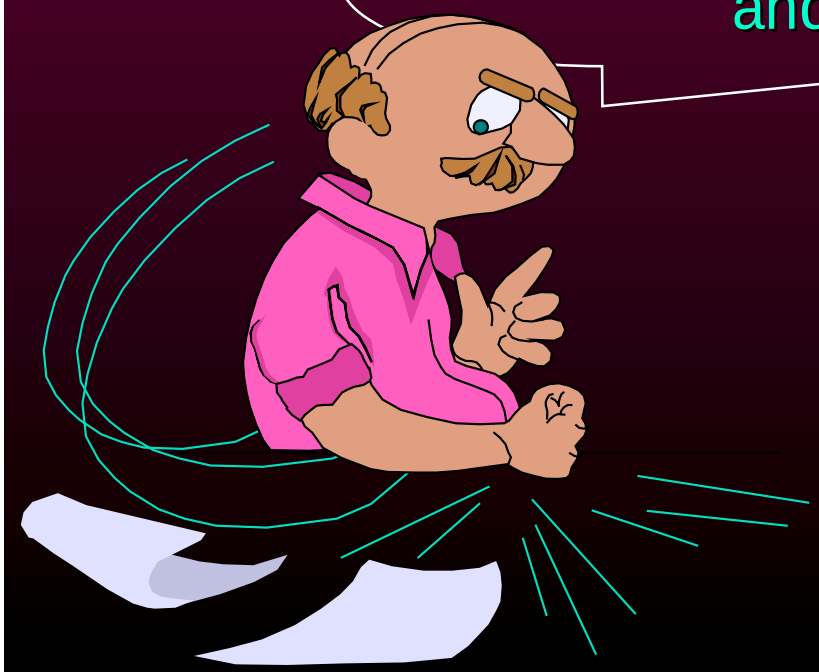
So, if the new VPI routines are harder to use, and most simulators do not support them,

Why transition to the new PLI standard?

You Don't Have a Choice

You MUST switch to the new VPI standard ! !

New features in the proposed 1998 Verilog HDL will only be supported in the VPI standard, Including support for complex intellectual property and deep submicron timing



When Should You Switch to VPI?

- Here's the dilemma:
  - You've got to switch to the new VPI PLI standard
  - Most Verilog simulators do not support the VPI routines

So When Should You Make the Transition?

- Suggestions for a smooth transition:
  1. Start learning the new VPI routines right now
     - It's not as easy as it might look
  2. For 1998 and 1999, write every PLI application twice
     - Using the older TF and ACC routines
     - Using the newer, more powerful VPI routines

A Prediction

Stu Sutherland predicts:

- By early 1999:
  - Any Verilog simulator that does not support the VPI standard will be a dying product
- By late 1999:
  - Any PLI application written in the older TF and ACC standard will be worthless