22805 SW 92nd Place, Tualatin, Oregon 97062 • Phone: (503) 692-0898 • FAX: (503) 692-1512 • Web: www.sutherland-hdl.com

# Transitioning to the New PLI Standard

Stuart Sutherland
Sutherland HDL, Inc., Portland, Oregon
www.sutherland-hdl.com

## 1.0 Abstract

*This paper explores the challenges that Verilog Programming Language Interface (PLI) application developers will face when transitioning from using the older TF and ACC PLI standard (also known as "PLI 1.0") to the new VPI PLI standard (also known as "PLI 2.0"). The older standard has been in use for nearly 15 years. It is well known, supported by all major Verilog simulators, and there are hundreds, possibly thousands, of commercial and proprietary applications written with the older PLI standard. The VPI library provides more capability, but is harder to use and is not supported by most major Verilog simulators. So why should users transition to the new standard? How hard will the transition be? And when is the right time to switch (if ever?)? This paper presents the advantages and disadvantages of both the old and new PLI standards. The issues with transitioning to the new PLI standard are presented and answers are suggested on if and when PLI application developers should transition the new standard.*

## 2.0 Introduction

The Verilog Programming Language Interface is a crucial and powerful part of the Verilog standard. It provides a means for Verilog users to extend the capabilities of Verilog simulators by allowing Verilog HDL models to call C programs. These C programs can interact with the Verilog simulation by reading the simulated design structure, reading simulation values, modifying simulation values, and scheduling simulation events. The Verilog Hardware Description Language was created to model hardware. The authors of the Verilog language purposely decided that it was not wise to try to build every conceivable and inconceivable capability into the Verilog HDL. Non-hardware modeling issues such as file I/O, foreign model import and co-simulation communication are best handled by the C programming language. Therefore, the authors of Verilog created a procedural interface, the Verilog PLI, to allow an open and virtually unlimited means of extending the capabilities of the hardware description language using the C programming language.

Even though the authors of Verilog provided for a procedural interface from the beginning, they did not envision all of the many creative ways users would dream up for using the PLI. From 1984 through 1990, Verilog was a proprietary language owned and controlled solely by Gateway Design Automation (Gateway merged with Cadence Design Systems late in 1989). In those early years, the PLI was not standardized. It evolved and changed with every release of the Gateway Verilog-XL product. There was no real specification for the PLI to evolve by, and one release of Verilog-XL would add features that the next release would remove or modify. In brief, the Verilog PLI in 1990 was very powerful and widely used, but the PLI was also in a mess.

## 3.0 The old PLI standard (“PLI 1.0”)

In 1990, Cadence released the Verilog HDL and PLI to the public domain. Open Verilog International (OVI) was formed to manage the new open Verilog standard and promote the use of Verilog. OVI labeled the 1990 PLI standard “PLI 1.0”. This version of the PLI was essentially what existed in the Cadence Verilog-XL simulator version 1.6a. The library of C functions provided in this version mostly begin with the letters “tf_” and “acc_”, and in the IEEE-1364 standard are referred to as TF and ACC functions.

**Advantages:**

Even now, in 1998, almost every commercial and proprietary PLI application is written using the TF and ACC functions of the “old” standard. There is a reason for this. The TF and ACC functions offer several advantages:

- The TF and ACC functions have been in use for more than 12 years.
- All major Verilog simulators support the TF and ACC functions
- Many experienced Verilog designers know how to write applications using the TF and ACC C functions
- Example PLI examples are readily available
- An extensive library of TF and ACC functions (over 200) make it easy to access specific data in a Verilog simulation
- Applications can be written quickly using a non-structured, straight-forward style — but don’t show the code to a programming professor :)

**Disadvantages**

There are also several disadvantages with using the old TF and ACC standard:

- The syntax and semantics is inconsistent for the more than 200 C functions in the PLI library.
- The relationship of Verilog objects and how to traverse a Verilog design structure using TF and ACC functions is not documented.
- The TF and ACC functions cannot access or control the execution of RTL models.
- The TF and ACC functions have very limited access to memory models.
- There are many undocumented “features” in the TF and ACC functions. Most major Verilog simulators have tried to clone Verilog-XL’s undocumented features, but the cloning is not always perfect.
- Cadence did not release many of the old C functions from earlier generations of Verilog-XL’s PLI. Cadence felt many of the old functions were obsolete, and should not be used, even though they still exist in the Cadence Verilog-XL simulator.
- After releasing the PLI to the public domain in 1990, Cadence has continued to evolve the PLI capabilities of Verilog-XL, adding new C functions which were not released to the public domain or Open Verilog International. Application developers who use these functions risk making their applications non-portable to other Verilog simulators.
- As an open standard, the TF and ACC functions are “frozen”. They are not being enhanced for current Verilog HDL language features, clone proprietary functions in Verilog-XL or implement new capabilities being added to the Verilog HDL.

## 4.0 The new VPI standard (“PLI 2.0”)

In 1993, Open Verilog International released a new PLI standard, which was called “PLI 2.0”. This new standard was designed to totally replace the old PLI, and correct all of the deficiencies of the old standard. Because PLI 2.0 was designed as a replacement, it was not backward compatible with the old PLI 1.0 standard. It was not possible, and not intended, to have old PLI applications running with new applications.

Also in 1993, OVI submitted a request to the IEEE to standardize the Verilog HDL and PLI. The IEEE-1364 Verilog standard working group debated whether to standardize the old and widely used

PLI 1.0 standard or the new and never used PLI 2.0 standard. The working group resolved the question by incorporating both OVI standards into a single IEEE-1364 standard. The terms “PLI 1.0” and “PLI 2.0” were replaced by “TF” “ACC” and “VPI” functions. The VPI functions are a re-write of OVI’s PLI 2.0, and provide complete backward compatibility with the older TF and ACC functions, which originate with OVI’s PLI 1.0 standard.

It is important to note that the VPI functions provide a nearly 100% duplication of the functionality of the TF and ACC functions. This redundancy is necessary in order to provide full PLI capability in the new PLI standard, and yet remain fully backward compatible with the old standard.

**Advantages**

The new VPI functions offer many advantages over the older TF and ACC functions:

- The VPI functions are written to a carefully planned specification.
- The relationship of Verilog objects and traversing Verilog design structures is thoroughly documented.
- The syntax and semantics of the VPI functions is consistent and simple.
- There are only 27 VPI functions, compared to over 200 TF and ACC functions.
- VPI functions provide complete access to everything in a Verilog simulation, including RTL statements, statement execution flow and memory models.
- VPI functions force a proper structured programming style.
- New Verilog HDL functionality in the proposed IEEE 1364-1998 standard will be supported by enhancements to the VPI functions.

**Disadvantages**

The VPI functions do have a few disadvantages:

- There a very few engineers who have experience writing VPI applications.
- It is much more difficult to learn to write VPI applications compared to TF and ACC applications. This is due to both the small number of VPI functions in the PLI library and because VPI functions require a more disciplined structured programming style.
- The VPI standard uses object diagrams to illustrate both Verilog object relationships and object

properties. Some engineers find the diagrams intuitive, but many have a difficult time using the diagrams.

- Only one major simulator vendor is currently supporting the complete TF, ACC and VPI standard; Cadence, with the Verilog-XL and NC-Verilog simulators.

## 5.0 Why stay with the old PLI standard

There are compelling reasons to continue to use the old TF and ACC functions in the PLI standard. First and foremost is that while all major Verilog simulators support the old standard, most do not support the new VPI functions, and most simulator vendors do not plan to support the VPI functions any time soon. Transitioning to the new PLI standard will shut the door on being able to use a PLI application in a variety of Verilog simulators. Non-portability is a caution flag for those developing proprietary in-house PLI applications; It means the developer needs to ensure that the simulators used at that company support the new PLI standard. For companies writing commercial PLI applications, non-portability is a large red stop sign. Commercial applications need to work with as many Verilog simulators as possible to be successful in the market.

The apparent lack of interest from simulator vendors to support the new PLI standard is the biggest reason to stay with the old PLI standard. There are other reasons to not transition to the new standard. Most companies have experience developing PLI applications using the old PLI standard. The learning curve to transition to the new standard is not trivial; the VPI functions are more difficult to use and require a more disciplined C programming style. Another important consideration is the robustness of the simulator implementation of the PLI standard. The older TF and ACC functions have existed for many years, and despite the annoying undocumented “features” of the standard, the implementations are tried and true. When simulator vendors do begin to implement the new VPI functions, there will likely be bugs and new undocumented features that will take one, two or even more releases of the simulator to fix. Implementations of the VPI standard will probably be very unstable for the next two years.

## 6.0 Why transition to the new PLI standard

There are equally compelling reasons to transition to the new PLI standard as soon as possible. Foremost among these reasons is that only the VPI functions are being updated in the IEEE standard to work with new enhancements to the Verilog HDL. The proposed 1998 Verilog HDL will include powerful module instance generation to meet the needs of configurable IP models. Design

verification will be greatly enhanced with constructs such as re-entrant tasks and recursive functions. New constructs are being added to increase the HDL timing accuracy for very deep submicron technologies, and the SDF delay back annotation standard is being enhanced to utilize the increased timing accuracy of the models. To take advantage of these new language features, PLI application developers will have to use the new PLI standard.

Another reason for transitioning to the new standard is that the VPI functions provide much more access to a Verilog simulation. This added access opens the door for many powerful commercial and proprietary PLI applications to be developed. The VPI functions also force a more disciplined C programming style, which may annoy hardware engineers who are trying to write C code, but results in more robust and maintainable PLI applications.

## 7.0 If you must transition, when should you make the switch?

It is not really a question of ***if*** PLI developers should transition to the new PLI standard. The added capabilities of the VPI functions and the fact that only the VPI functions are being updated to work with enhancements to the Verilog HDL will force every developer to use the new PLI standard. The real question is ***when*** application developers should transition to the new PLI standard. This is a very difficult question to answer, and will be dictated in great measure by the target audience of the PLI application.

Proprietary in-house PLI applications can transition today if the Verilog simulators that need the applications are the Cadence Verilog-XL and the Cadence NC-Verilog. Other simulators might also be used within the company, as long as they do not need the PLI applications written with the new PLI standard. But developers of in-house PLI applications need to carefully look at the long term usage of the PLI application. Using the new PLI standard today may limit the company's choices of Verilog simulators for the rest of this century.

Commercial PLI applications that are intended to sell to a variety of Verilog users cannot transition to the new PLI standard today. Commercial applications are stuck using the old PLI standard until more simulator vendors decide to support the VPI standard, and can provide robust, tested implementations of the VPI functions. This is a severe trade-off. There are many unique and powerful features in the VPI functions, such as the ability to trace the execution of RTL statements, that could make new commercial applications stand head and shoulders above today's commercial applications.

Proprietary ASIC delay calculators have no choice but to transition to the new PLI standard

immediately. The new and accurate timing enhancements in the proposed 1998 Verilog HDL can only be accessed and annotated with the VPI functions. Delay calculators written with the old PLI standard will not work with new ASIC libraries. An alternative is to drop proprietary delay calculation using the PLI and instead use static delay calculation with SDF back annotation files. But this presupposes that simulator vendors will support the proposed SDF changes required for the enhanced timing in the Verilog HDL. Since most SDF annotators are implemented using the PLI, the annotators must first be re-written using the new PLI standard.

## 8.0 Tips on reducing the transition headaches

This paper has emphasized the need to eventually transition to the new PLI standard, as well as the many issues that can (and will) arise with making the transition. There may be little that can be done to avoid the headaches of the transition, but there are pro-active steps that can be taken to reduce the pain involved. Here are a few suggestions:

- Learn the new PLI standard now. You may not be able to transition immediately, but by understanding the C programming requirements of the VPI functions, you can write applications in the old PLI standard that will easily port to the new standard in the future. Most PLI applications are written by hardware engineers who have little concept—and a strong aversion to—the structured programming style VPI functions enforce. Now is the time to learn how to use structured programming in PLI applications, and why it is important.

- Consider writing two versions of smaller, in-house PLI applications. One using the older PLI standard and one using the new standard. This requires an up front time investment, but provides a tested and working application in the new standard that can be used with each simulator that adds support for the new standard. And if a simulator does not work correctly with a known good application, then it is much easier to show that there is a bug in the simulator’s implementation of the VPI functions.

- As a PLI application developer, start making a lot of noise with the your simulator vendors for support for the new PLI standard. And make sure your demands are heard over and over again. Most of the major simulator vendors have publicly stated that they will not add support for the new PLI standard until enough of their customers are demanding it (the author of this paper could write an entire essay on his opinion of this “wait-and-see” attitude of many simulator vendors).

## 9.0 Conclusion

The new Verilog PLI standard offers compelling reasons to transition to the standard. But the transition is full of perils and pitfalls. There is a difficult learning curve. In the short term, the standard is not supported by most Verilog simulators, making portability a major issue. As simulators do implement the new standard, there will likely be bugs and “features” to deal with.

At some point in time, every PLI application developer will need to transition to the new PLI standard. However, each company will need to carefully consider when to make this transition. The decision factors include when are the unique capabilities of the new standard really needed, and how important is application portability across multiple Verilog simulators. The decisions will involve compromises and trade-offs that cannot be avoided.

PLI applications developers can take pro-active steps to reduce the difficulties of transitioning to the new standard. These steps include learning the new standard as soon as possible and continually requesting that your simulator vendors support the new PLI standard.

## 10.0 About the author

Stuart Sutherland is the founder of Sutherland HDL Consulting, located in Portland, Oregon. He holds a B.S. in Computer Science with an emphasis in Electronic Engineering. Stuart has more than 12 years of experience in hardware design and over eight years experience with Verilog, including working as a senior applications engineer for Gateway Design Automation, the company that founded Verilog. Stuart became an independent Verilog consultant in 1992, and provides expert Verilog design services and Verilog language and PLI training. He is actively involved in the IEEE Verilog standardization committee, and is the co-chair of the IEEE-1364 PLI task force as well as technical editor for the PLI sections of the IEEE 1364 Verilog Language Reference Manual. Stuart is the author of the popular “Verilog HDL Quick Reference Guide” and “Verilog PLI Quick Reference Guide”, and is currently writing a comprehensive text book on the PLI. Stuart can be contacted at stuart@sutherland-hdl.com.