# Improving Simulation Performance Using PLI



## 1 Introduction

VCS allows you to enable access (ACC) capabilities throughout your design and specify ACC capabilities for specific parts of your design. It also enables you to specify selected ACC capabilities using a PLI tab file. Enabling ACC capabilities impacts performance, VCS allows you to enable only the ACC capabilities that you need.

PLI is the Programming Language Interface (PLI) between C/C++ functions and VCS. It helps to link applications containing C/C++ functions with VCS, so that they execute concurrently. VCS supports PLI 1.0 and PLI 2.0 routines for the PLI. Therefore, you can use VPI and ACC routines to write the PLI application.

Access capabilities (read, write, force, and callback) can be enabled by the PLI application using the following methods:

- Module-Based PLI
- Instance-Based or Signal-Based PLI

This document describes the methods to enable the access capabilities to improve the compile/runtime performance.

## 2 Module-Based PLI

This method enables the module-level ACC capability. This capability is applied to the complete design wherever the modules are connected. Therefore, this capability is also applied to the unwanted design hierarchies and affects the compile/runtime performance.
To improve the performance in module-based PLI, you can restrict the ACC capabilities using learn PLI.

### 2.1 Enabling Module Level PLI Learn

Use the `+vcs+learn+pli` runtime option to instruct VCS to learn the access capabilities (signal at module level) that are used by the modules in your design and write them into a secondary PLI tab file named `pli_learn.tab`. You can use this tab file to recompile your design so that subsequent simulations use only the ACC capabilities that are required.

Recompile your design using the `+applylearn` compile-time option and run the simulation.
By default, VCS selects the `pli_learn.tab` file.

**Use Model**

```
% cat pli.tab
   acc+=frc:dut //Enable force capability on signals/ports of "dut" module.

% vcs -P pli.tab <design file>
% . /simv +vcs+learn+pli //learn the debug capability on module-level PLI
% vcs -P pli.tab +applylearn+pli_learn.tab <design File>
```

## 2.2 Enabling Signal PLI Learn at Instance Level

Use the `+vcs+learn+pli+sigpli +vcs+learn+pli+instpli` runtime option to instruct VCS to learn the access capabilities (signal at instance level) that are used in the simulation and write them into a secondary PLI tab file named, `pli_learn.tab`.

Recompile your design using the `+applylearn -instpli -sigpli` compile-time option and then run the simulation. By default, VCS selects the `pli_learn.tab` file.

# 3 Instance-Based or Signal-Based PLI

ACC capabilities are required only for few signals in a module. Enabling ACC capabilities on the entire module creates a significant performance overhead that can be eliminated using the instance/signal PLI (`instpli/sigpli`) constructs.

The `instpli/sigpli` constructs allow ACC capabilities to be specified for module signals instead of entire modules. This allows VCS to learn the capability by instance/signal based PLI.

**Use Model**

```
% cat pli.tab
  acc+=frc:top.dut.a // Enable force capability on the signals of the
"top.dut.a" module.

% vcs -P pli.tab <design file> -sigpli -instpli // Enable debug capability on
a signal at instance level
%. /simv
```

# Copyright Notice and Proprietary Information



## Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

## Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at http://www.synopsys.com/Company/Pages/Trademarks.aspx.
All other product or company names may be trademarks of their respective owners.

## Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

## Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com