Beacon-PSL

Comprehensive Test Suite for PSL Simple Subset

Key Advantages

- Backed by Interra's field-proven expertise in developing HDL-based test suites for VHDL and Verilog
- Developed in partnership with significant EDA majors
- Conforming to accepted definition and interpretation of the language
- Providing an unbiased quality analysis of EDA tools
- Complete coverage of constructs and styles
- Comprehensive validation of syntax and simulation semantics

Highlights

- Over 2,000 test cases along with test benches
- Test cases in both Verilog and VHDL for validating PSL Simple subset
- Well organized test cases highlighting testing objectives
- Detailed test plans with cross-reference to test cases
- Golden output for comparison

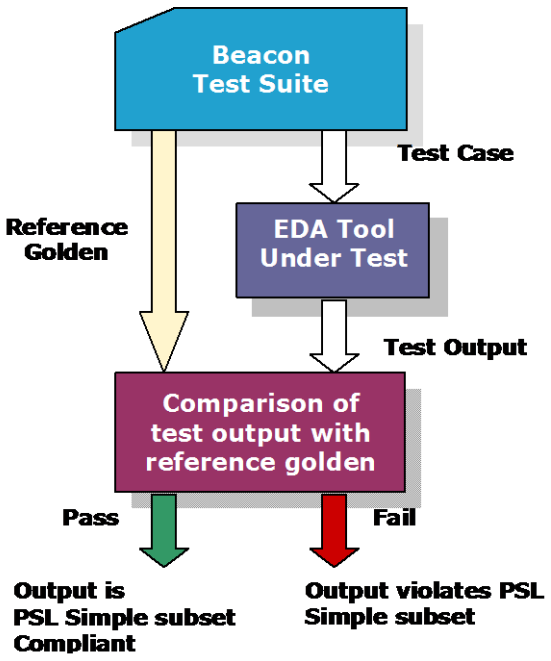
Addressing the needs of EDA tool developers to quickly evaluate the quality of their products, Interra's Beacon-PSL delivers a comprehensive test suite based on PSL (IEEE 1850-2005) standards.

You can use Beacon-PSL to validate PSL support in EDA tools. In addition, you can characterize EDA tools for coverage and quality across PSL constructs.

Enabling you to discover language non-compliance early in the product development and testing life cycle, Beacon-PSL offers:

- Reduced development costs and time-to-market EDA products
- Development of standard-based products
- Precise evaluation of bugs and errors in the product
- Measure of product quality
- Unbiased feedback on product quality
- Regression tests for quality assurance

The test cases and test benches can be applied to the EDA tool under evaluation and results can be compared with golden reference that is provided with Beacon-PSL.



The Beacon-PSL Features

Comprehensive Test Suite

Covers PSL, IEEE std. 1850-2005. The test cases are well organized and enable you to evaluate syntax, semantics, and simulation aspects of a PSL-based tool.

Syntax and Semantics Cover

Beacon-PSL includes over 2,000 test cases that cover constructs as specified in the table below. Along with positive test cases, there are negative test cases that validate correct behavior by taking in erroneous inputs.

Positive test cases include Verilog /VHDL test cases, test bench, and expected output. Negative test cases include expected compiler output messages.

Language Construct	Number of Test Cases
FL Property	579
Implication Operator	71
Logical FL Property	36
LTL Operator	100
Macros	56
Modeling Layer	21
Replicated Properties	52
Named Endpoint	87
Named Sequence	87
Built-In Function	149
Compound SERE	96
Simple SERE	26
Union	5
PSL Meta Comment	18
Sequence	247
Verification Directives	147
Verification Units	136
HDL Dependencies	67
Designs with mixed constructs	73
Total	**2053**

Simulation Cover

This test suite provides exhaustive test cases to cover supported and unsupported styles and constructs of PSL in Verilog/VHDL flavors. Positive test cases, with test benches and expected output, validate simulator and equivalent tools.

Well Documented Test Plans

The test plans describe all test objectives and are categorized by sections. Each test case has a reference to the section number of the test plan.

Test Benches and Reference Golden

Provides test benches to instantiate test cases and apply vectors on inputs. Outputs are captured after an appropriate interval and written on to a file. Reference golden output is also provided for these test benches for all the test cases. You can easily apply the test bench to the test tool and compare the outputs.

Sample Verilog Test Case

```
--** Purpose:      To test the syntax of SERE
                   consecutive repetition
                   operator.
--** LRM :         Sections 6.1.1.1.3
--** TestPlan:     Sections 3.1.1
--** Kind :        Syntax/Semantics
--** Status:       ANALYSIS_SHOULD_PASS

*****************************************

module top(out,clk,reset);
 parameter width=3'd4;
 input clk,reset;
 output [width-1:0] out;
 reg [width-1:0]    out;

 /********* Assertions*********/

 // psl default clock = (posedge clk);
 // psl assert always
 {out[0];!out[0];out[0][*3]};

 always@(posedge clk)
 begin
   if(!reset)
    out<=0;
   else
    out<={~out[0],out[width-1:1]};
 end
endmodule
```

Interra Systems, Inc.
1601 S. De Anza Boulevard, Suite 212, Cupertino, CA 95014
Phone: +1 408 579 2000, Fax: +1 408 579 2050
Email: info@interradesign.com Web: www.interradesign.com