LEARNING system-verilog

#system-verilog

Table of Contents

About

Chapter 1: Getting started with system-verilog

Remarks

SystemVerilog is the successor language to Verilog. Originally created by Accellera as an extension language to Verilog IEEE Std 1364-2001, SystemVerilog was accepted as an IEEE standard in 2005. In 2009, IEEE merged Verilog (IEEE 1364) into SystemVerilog (IEEE 1800) as a unified language. Like its predecessor, SystemVerilog is supported by many FPGA (Field Programmable Gate Array) vendors and ASIC (Application Specific Integrated Circuit) tool vendors. SystemVerilog was created to enhance HDL design development and has dedicated features for verification.

SystemVerilog consists of 3 main sub-languages:

- Design directives : Allows designers to write RTL more concise, explicit, and flags mistakes traditionally not found until synthesis.
- Object oriented classes : Used for verification, allows test-bench code to be more flexible and reusable. This capability spurred the creation of verification methodologies: OVM, VMM, UVM
- Assertions : Used for verification and coverage of protocols and internal sequential signals.

Versions

Version	Release Date
SystemVerilog IEEE Std 1800-2012	2013-02-21
SystemVerilog IEEE Std 1800-2009	2009-12-11
SystemVerilog IEEE Std 1800-2005	2005-11-22

Examples

Installation or Setup

In order to compile and run SystemVerilog code a tool called a simulator is needed. Most commonly, commercial tools from one of the Big Three EDA companies is used:

- Cadence Incisive
- Mentor Graphics QuestaSim
- Synopsys VCS

Other EDA vendors also provide simulators:

- Aldec Riviera-PRO
- Xilinx Vivado

Free and open source tools also exist, that support different subsets of the LRM:

- Verilator

Hello world

```
// File 'test.sv'

// Top module that gets instantiated automatically when simulation is started
module test;

  // Thread gets started at the beginning of the simulation
  initial begin

    // Call to system task to print output in simulator console
    $display("Hello world!");
  end

endmodule
```

Running in Cadence Incisive:

```
irun test.sv
```

Read Getting started with system-verilog online: https://riptutorial.com/system-verilog/topic/2829/getting-started-with-system-verilog

Credits

S. No	Chapters	Contributors
1	Getting started with system-verilog	AndresM, Community, Greg, Qiu, ScottJ, Tudor Timi