



EBook Gratis

APRENDIZAJE system-verilog

Free unaffiliated eBook created from
Stack Overflow contributors.

#system-
verilog

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con el sistema-verilog.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración.....	2
Hola Mundo.....	3
Creditos.....	4

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [system-verilog](#)

It is an unofficial and free system-verilog ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official system-verilog.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Comenzando con el sistema-verilog

Observaciones

SystemVerilog es el idioma sucesor de [Verilog](#) . Originalmente creado por Accellera como un lenguaje de extensión para Verilog [IEEE Std 1364-2001](#) , SystemVerilog fue aceptado como un estándar IEEE en 2005. En 2009, IEEE fusionó Verilog (IEEE 1364) en SystemVerilog (IEEE 1800) como un idioma unificado. Al igual que su predecesor, SystemVerilog es compatible con muchos proveedores de FPGA (Field Programmable Gate Array) y proveedores de herramientas de ASIC (Circuito integrado específico de la aplicación). SystemVerilog fue creado para mejorar el desarrollo del diseño HDL y tiene características dedicadas para la verificación.

SystemVerilog consta de 3 sub-idiomas principales:

- Directivas de diseño: permite a los diseñadores escribir RTL de forma más concisa, explícita y señala los errores que tradicionalmente no se encuentran hasta la síntesis.
- Clases orientadas a objetos: utilizadas para verificación, permiten que el código de banco de pruebas sea más flexible y reutilizable. Esta capacidad estimuló la creación de metodologías de verificación: [OVM](#) , [VMM](#) , [UVM](#).
- Afirmaciones: se utilizan para la verificación y cobertura de protocolos y señales secuenciales internas.

Versiones

Versión	Fecha de lanzamiento
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

Instalación o configuración

Para compilar y ejecutar el código SystemVerilog se necesita una herramienta llamada simulador. Más comúnmente, las herramientas comerciales de una de las compañías de Big Three EDA se utilizan:

- Cadencia incisiva

- Mentor Graphics QuestaSim
- Synopsys VCS

Otros proveedores de EDA también ofrecen simuladores:

- Aldec Riviera-PRO
- Xilinx Vivado

También existen herramientas gratuitas y de código abierto, que admiten diferentes subconjuntos del LRM:

- Verilator

Hola Mundo

```
// 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
```

Corriendo en cadencia incisiva:

```
irun test.sv
```

Lea Comenzando con el sistema-verilog en línea: <https://riptutorial.com/es/system-verilog/topic/2829/comenzando-con-el-sistema-verilog>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con el sistema-verilog	AndresM , Community , Greg , Qiu , ScottJ , Tudor Timi