



**EBook Gratuito**

# APPENDIMENTO

---

# system-verilog

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#system-  
verilog

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con system-verilog.....</b>	<b>2</b>
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Ciao mondo.....	3
<b>Titoli di coda.....</b>	<b>4</b>

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](mailto:info@zzzprojects.com)

# Capitolo 1: Iniziare con system-verilog

## Osservazioni

SystemVerilog è il linguaggio successivo a [Verilog](#). Originariamente creato da Accellera come linguaggio di estensione per Verilog [IEEE Std 1364-2001](#), SystemVerilog è stato accettato come standard IEEE nel 2005. Nel 2009, IEEE ha unito Verilog (IEEE 1364) in SystemVerilog (IEEE 1800) come lingua unificata. Come il suo predecessore, SystemVerilog è supportato da molti fornitori di FPGA (Field Programmable Gate Array) e ASIC (Application Specific Integrated Circuit). SystemVerilog è stato creato per migliorare lo sviluppo del design HDL e ha funzioni dedicate per la verifica.

SystemVerilog è costituito da 3 lingue secondarie principali:

- Direttive di progettazione: consente ai progettisti di scrivere RTL in modo più conciso, esplicito e segnala errori tradizionalmente non rilevati fino alla sintesi.
- Classi orientate agli oggetti: utilizzate per la verifica, consentono al codice del banco di prova di essere più flessibile e riutilizzabile. Questa capacità ha stimolato la creazione di metodologie di verifica: [OVM](#), [VMM](#), [UVM](#)
- Asserzioni: utilizzate per la verifica e la copertura di protocolli e segnali sequenziali interni.

## Versioni

Versione	Data di rilascio
SystemVerilog <a href="#">IEEE Std 1800-2012</a>	2013/02/21
SystemVerilog <a href="#">IEEE Std 1800-2009</a>	2009-12-11
SystemVerilog <a href="#">IEEE Std 1800-2005</a>	2005-11-22

## Examples

### Installazione o configurazione

Per compilare ed eseguire il codice SystemVerilog è necessario uno strumento chiamato simulatore. Più comunemente, vengono utilizzati strumenti commerciali di una delle tre grandi società EDA:

- Cadenza incisiva
- Mentor Graphics ThisSim
- Synopsys VCS

Altri venditori EDA forniscono anche simulatori:

- Aldec Riviera-PRO
- Xilinx Vivado

Esistono anche strumenti gratuiti e open source che supportano diversi sottoinsiemi di LRM:

- Verilator

## Ciao mondo

```
// 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
```

Esecuzione in cadenza incisiva:

```
irun test.sv
```

Leggi Iniziare con system-verilog online: <https://riptutorial.com/it/system-verilog/topic/2829/iniziare-con-system-verilog>

---

# Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con system-verilog	<a href="#">AndresM</a> , <a href="#">Community</a> , <a href="#">Greg</a> , <a href="#">Qiu</a> , <a href="#">ScottJ</a> , <a href="#">Tudor Timi</a>