



**SystemVerilog:
Right Here! Right Now!**



Agenda

- **Welcome:**
 - Gabe Moretti, Technical Editor, EDA and ASICs, EDN
- **SystemVerilog Right Here! The Handoff to the IEEE**
 - Dennis Brophy, Chairman, Accellera & Edward Rashba, Manager, New Technical Programs, IEEE Standards Association
- **SystemVerilog 3.1a Unwrapped**
 - Vassilios Gerousis, Chairman, Accellera Technical Committee
- **SystemVerilog in Action: A User's Perspective:**
 - Matt Maidment, Sr. CAD Engineer, Intel Corporation
- SystemVerilog Right Now! Rapid fire support announcements from your SystemVerilog suppliers (*refer to separate presentation*)
- Closing and Prize Drawing



SystemVerilog Right Here! The Handoff to the IEEE

Dennis Brophy

Chairman
Accellera

Edward Rashba

Manager
New Technical Programs
IEEE Standards Association

Accellera Standardization Process

- Accellera has a mature standardization process:
 - Level0: Identify needs through users and vendors
 - Level1: Users and Vendor supported Incubation
 - Level2: Refinement and Solidification through usage and tool implementation
 - Level3: Distribute standards through IEEE to IEC

Accellera Policy & IEEE

- Accellera policy is to offer its standards to IEEE
 - When standard is solid through Accellera process
 - Sponsor the creation of committees and help in the development of IEEE standards
 - This process started with OVI, and continues developing standard with IEEE process
 - Accomplishment In IEEE/IEC: Verilog, SDF, PDEF, SPEF, ALF, Verilog Synth. subset



SystemVerilog



Improving Accellera / IEEE Efficiency

- Engage directly with IEEE-SA
 - Improved efficiency
 - Impedance matched with Accellera
 - Company participation and voting rules
 - Direct support from IEEE during development process
 - Commercial backing in advance of standard
 - Pro-consumer to offer standards in a timely fashion
 - Industry relevant to address current market needs
 - Participate in shaping evolving process to develop and support international standards
 - Drive IEEE changes to support emerging standardization methods (Accellera OVL open-source, and others)

SystemVerilog 3.1a Handoff

- SystemVerilog 3.1a offered to IEEE P1800



**SystemVerilog 3.1a
Language Reference Manual**

Accellera's Extensions to Verilog®

Abstract: a set of extensions to the IEEE 1364-2001 Verilog Hardware Description Language to aid in the creation and verification of abstract architectural level models



The SystemVerilog logo, featuring the text "SystemVerilog" in a serif font, with a white swoosh underline that starts under "System" and ends under "Verilog".

The Accellera logo, consisting of the word "accellera" in a lowercase, italicized sans-serif font, enclosed within a blue oval shape.

SystemVerilog Path Through IEEE Standardization process

THANK YOU ACCELLERA!

Major Process Steps:

PAR Approval: June 23rd

Project Development: 10-16 months

Balloting & RevCom Submittal: 2 months

Approval: by end of 2005

The logo for SystemVerilog, featuring the text "SystemVerilog" in a white serif font, with a white swoosh underline that starts under "System" and ends under "Verilog".

The logo for accellera, featuring the word "accellera" in a blue, lowercase, sans-serif font, enclosed within a blue oval border.

IEEE Individual & Entity Model

- Individual Model
- “5 pillars” of consensus process
- IEEE-SA Member to ballot
- “One individual, one vote” on WG

Entity Model

- “5 pillars” of consensus process
- IEEE-SA Corporate Member to ballot
- “One company, one vote”

... Entity model enables industry-supported standards to move in time with the market

IEEE Value Proposition- Corporate

- Quick initiation and development
- Clearly defined rules
- “One company, One vote”
- Paths to international arena
- Stability and clout of IEEE standards
- Competitive cost structure
- Electronic and global distribution
- Experienced and professional support

IEEE-SA delivers...speed, legitimacy, global reach

SystemVerilog

accellera

Internationalization- IEC/IEEE Dual Logo Agreement

SystemVerilog to follow other international EDA successes:

- IEEE Std 1076™-2002, IEEE Standard VHDL Language Reference Manual
- IEEE Std 1076.4™-2000, IEEE Standard VITAL ASIC (Application Specific Integrated Circuit) Modeling Specification
- IEEE Std 1364™-2001, IEEE Standard Verilog Hardware Description Language
- IEEE Std 1497™-2001, IEEE Standard for Standard Delay Format (SDF) for the Electronic Design Process

SystemVerilog

INTERNATIONAL
STANDARD

IEC
601800

IEEE 1800

Standard digital interface for programmable instrumentation –

Part 2:
Codes, formats, protocols
and common commands



Reference number:
IEC 601800-2(C):2004
IEEE Std. 1800-2(C):1992



Thank You!

Dennis Brophy

**You know how to reach
me**

Edward Rashba

Tel. +1 732 465 6449

e.rashba@ieee.org

SystemVerilog

accellera



SystemVerilog 3.1a Update

Vassilios Gerousis
Accellera SystemVerilog Chair
Infineon Technologies



Outline

- Goals of SystemVerilog 3.1a
- Efforts
- Highlights Of SV 3.1a
- Summary

Timeline

- SystemVerilog 3.1A has gone through the last maturity step of Accellera standardization process
- Development of SystemVerilog 3.1A started In August of 2003 and was unanimously approved by the technical committees on March of 2004
- Accellera Board Approved 3.1A as SystemVerilog standard on April 29, 2004.

SystemVerilog 3.1A Goals

- Address issues exposed in SystemVerilog 3.0 / 3.1 standard through Implementation and Usage. Complete SystemVerilog 3.1A release by March/April 2004.
- Address errata in SystemVerilog 3.1 standard.
- Add minor enhancements: All enhancements are users driven (Sun, Intel, Motorola, Infineon, etc.)
- 100% compatibility with IEEE 1364 Verilog 2001 standards: Common membership with 1364 errata committee aided in the synchronization with the Errata Development of 2001 Verilog.

Update On Effort

- *Five companies donations*: Bluespec, Mentor Graphics, Motorola, Novas System and Synopsys donated to 3.1A.
 - Real Intent donated in 3.0 and 3.1
- Total BNF re-design of SystemVerilog (Superset of Verilog). Donated effort to IEEE 1364.
- LRM development took 4.5 man years of efforts.
- More than 200 experts.

SystemVerilog 3.1A Highlights

- Focus on tighter integration of SystemVerilog Assertion with all SV components (Design, Testbench, Interface).
 - SVA is targeted for designers and not just verification engineers.
 - Emphasis on “integrated design and verification” methodology.
- Address Tool Implementers and Users Feedback
 - More than 200 change requests (Errata) were addressed and approved.
 - Added user-requested enhancements through donations/proposals.

Technology Development

SystemVerilog Component	Number of Errata	Number of enhancements
Verification Layer (Testbench)	85	11
Design Layer	177	0
Assertion Layer	6	19
Interface Layer	7	4

Design Layer

- Advanced design enhancement
 - Memory system tasks for complex memory modeling
 - Operator overloading for simplified expressions
 - Tagged unions with pattern matching for code conciseness & improved formal analysis

Assertion Layer

- Assertion enhancements
 - Environmental constraints to facilitate formal analysis & random simulation (assume construct with randomization)
- Broaden the scope of assertion usage within SystemVerilog
 - (e.g. Use in function, added parameters to assertions, gated clock support)
- Enhancing the expressiveness of the language (e.g. local variables)

Verification

- Testbench enhancements
 - Fine-grain process control for multi-threaded testbench development
 - Dynamic & static queues and stream generation for complex verification scenarios
 - Virtual interfaces for flexibility, testbench reuse and expressiveness of testbench infrastructure
 - Random weighted case and functional coverage to support constrained-random verification methods

Interface Layer

- VPI Interface: Provide C/C++ API functions to SystemVerilog
 - Extended VPI model can access all capabilities of SystemVerilog.
 - New extensions to VPI to give full access to waveform files regardless of file format
- Some of the APIs that was completed in previous releases
 - Coverage VPI
 - Assertion VPI
 - Direct Programming Interface (DPI)

Improved Use Model

- Separate compilation
- Vendor-independent API to access proprietary waveform file formats for higher performance
- SystemVerilog tasks exported in the DPI so foreign language can interact with SystemVerilog as if it were interacting with its own. (C/C++ routine can consume time and block until a task completes)

Summary

- SystemVerilog 3.1A developed by 40 world wide experts with over 200 people on the email reflector.
- SystemVerilog is ready to use right now.
 - It has gone through three phases and it is solid and mature technology.
- SystemVerilog 3.1a LRM is available for free download
 - http://www.eda.org/sv/SystemVerilog_3.1a.pdf



SystemVerilog for Design

Matt Maidment
Intel Corporation



Motivation for SystemVerilog

- Moore's Law continues
- Must improve design productivity to keep pace
- My SystemVerilog participation focused on influencing development of a language that is equipped to address future design challenges
- Thanks to the efforts of all SystemVerilog committees, the language now enables designers to meet these challenges with:
 - Better Code, More Reuse & More Intent

Better Code

- Data-organization is key to readable, verifiable & debuggable code
- Variables, structs, arrays, unions & enums
 - Lead to more descriptive, readable code
 - Capture data relationships
 - Enable data movement with fewer lines of code
 - No more shifting reg's to wire's and vice versa
- void functions & always_comb encapsulate functionality
- Assertions capture negative space

More Reuse

- Verilog enables modular logic specification
- SystemVerilog adds modular interconnect specification
 - Structs & arrays bundle unidirectional data flow
 - Interfaces bundle bi-directional data flow
 - Data types and directions vary
 - Capture behavioral assertions once instead of distributed across the system
- Packages facilitate type sharing

More Intent

- Why spend simulation cycles to discover
 - A latch or flop isn't a latch or flop
 - always_latch, always_ff
 - Combinational logic has latches
 - always_comb
 - Net incorrectly has more than one driver
 - variables & always_*, assign, output
 - Simulation and synthesis disagree
 - unique/priority if/case
- Assertions capture
 - Boundary assumptions & expected behavior

The SystemVerilog Experience

- Users are heard through the Accellera SV process.
- Productive Integration of
 - Significant technology
 - Contributors throughout the industry
- Result is a language that is better equipped to meet today's design challenges and future opportunities

SystemVerilog

