SMASH: a Verilog-A simulator for analog designers

Gilles DEPEYROT,
Frédéric POULLET & Benoît DUMAS
DOLPHIN Integration

Outline

- Context & Goals
- Coding Guidelines
- Benchmark of Verilog-A vs. SPICE
  - Progress during the last year
  - Verilog-A limitations in the current implementations
  - Verilog-A vs. SPICE
- Conclusion
  - Status, Perspectives & Outlook

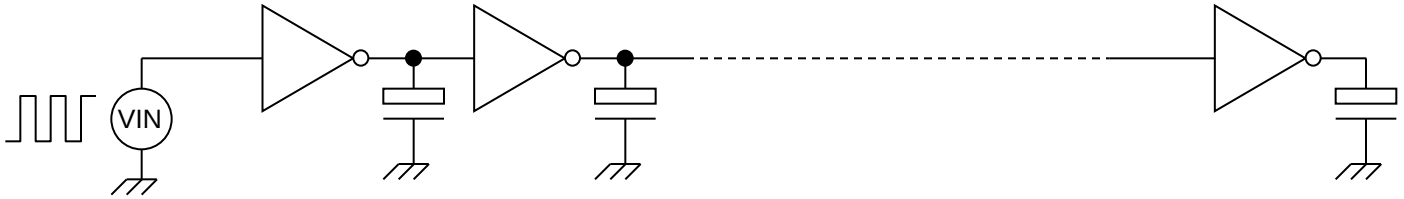
Context & Goals

- What needs to be done so that Verilog-A can become the standard for CM coding?
  - Comparing the “general purpose” Verilog-A compilers with the integrated SPICE devices
    - Detailed technical investigation of Verilog-A compilation aspects, not only for Compact Models
  - Benchmarking performed to understand current status and SMASH progress
    - Some results previously presented at MOS-AK in Frankfurt and Athens
    - Guidelines put together for CM coding
- What is at stake?
  - Fully taking into account SPICE-like integration of Verilog Compact Models in the ecosystem
  - Providing a viable and open alternative to “controlled” initiatives (such as TMI)

Coding Guidelines

- Writing compact models
  - Geoffrey J. Coram, “HOWTO (AND HOWNOT TO) WRITE A COMPACT MODEL IN VERILOG-A”, BMAS 2004
    www.bmas-conf.org/2004/papers/bmas04-coram.pdf
  - L. Lemaitre, C. Mc Andrew and W. Grabinski, “Standardization of Compact Device Modeling in High Level Description Language”, Nanotech 2003 Vol. 2
    http://www.nsti.org/publications/Nanotech/2003/pdf/X2402.pdf
- Optimizing compact models
  - “Guidelines for Verilog-A Compact Model Coding”, Nanotech 2010 Vol. 2
    www.techconnectworld.com/Microtech2010/a.html?i=628
  - “Verilog-A Compact Model Coding Whitepaper”
    www.dolphin-integration.com

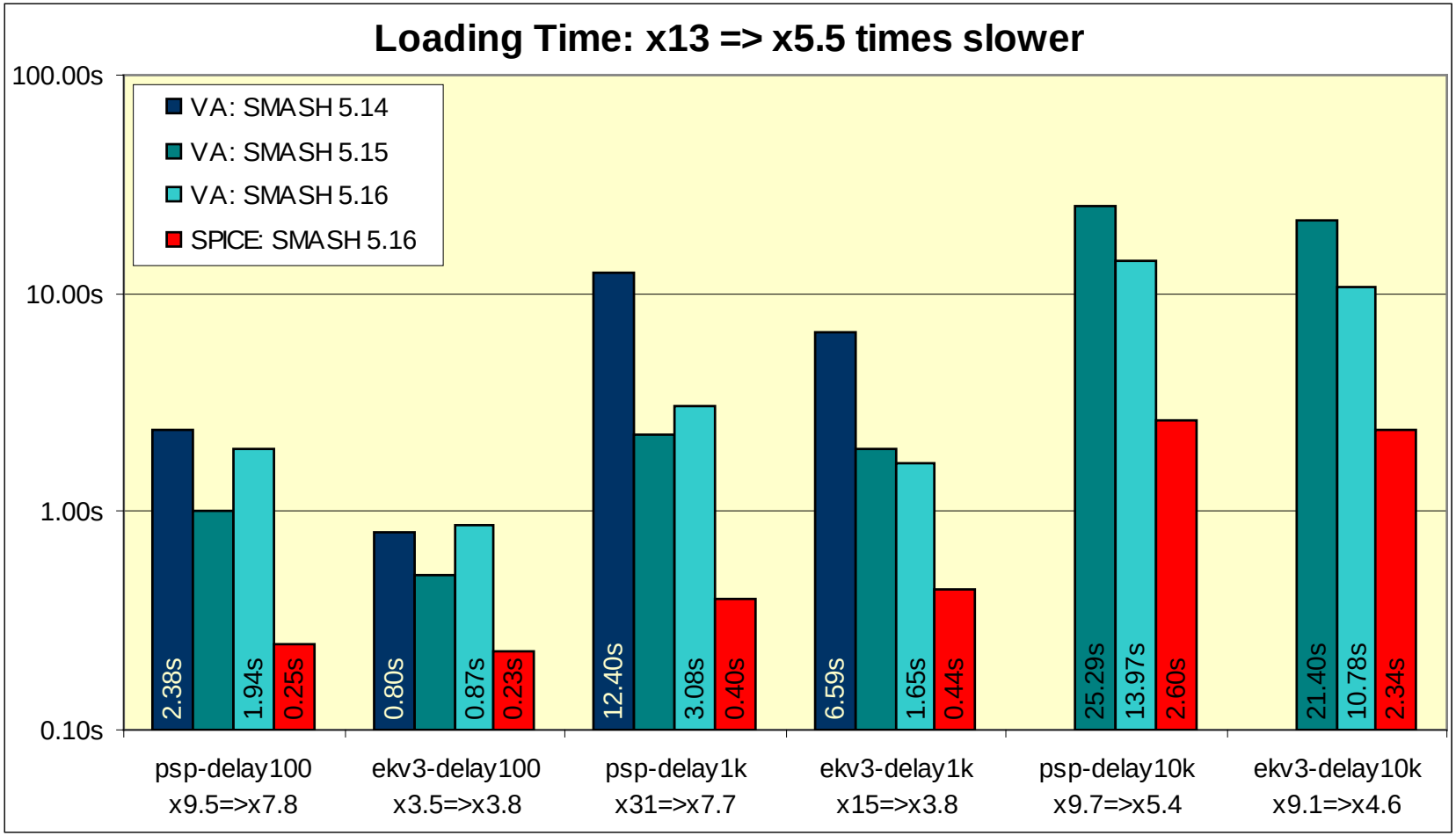
Benchmark of Verilog-A vs. SPICE Conditions



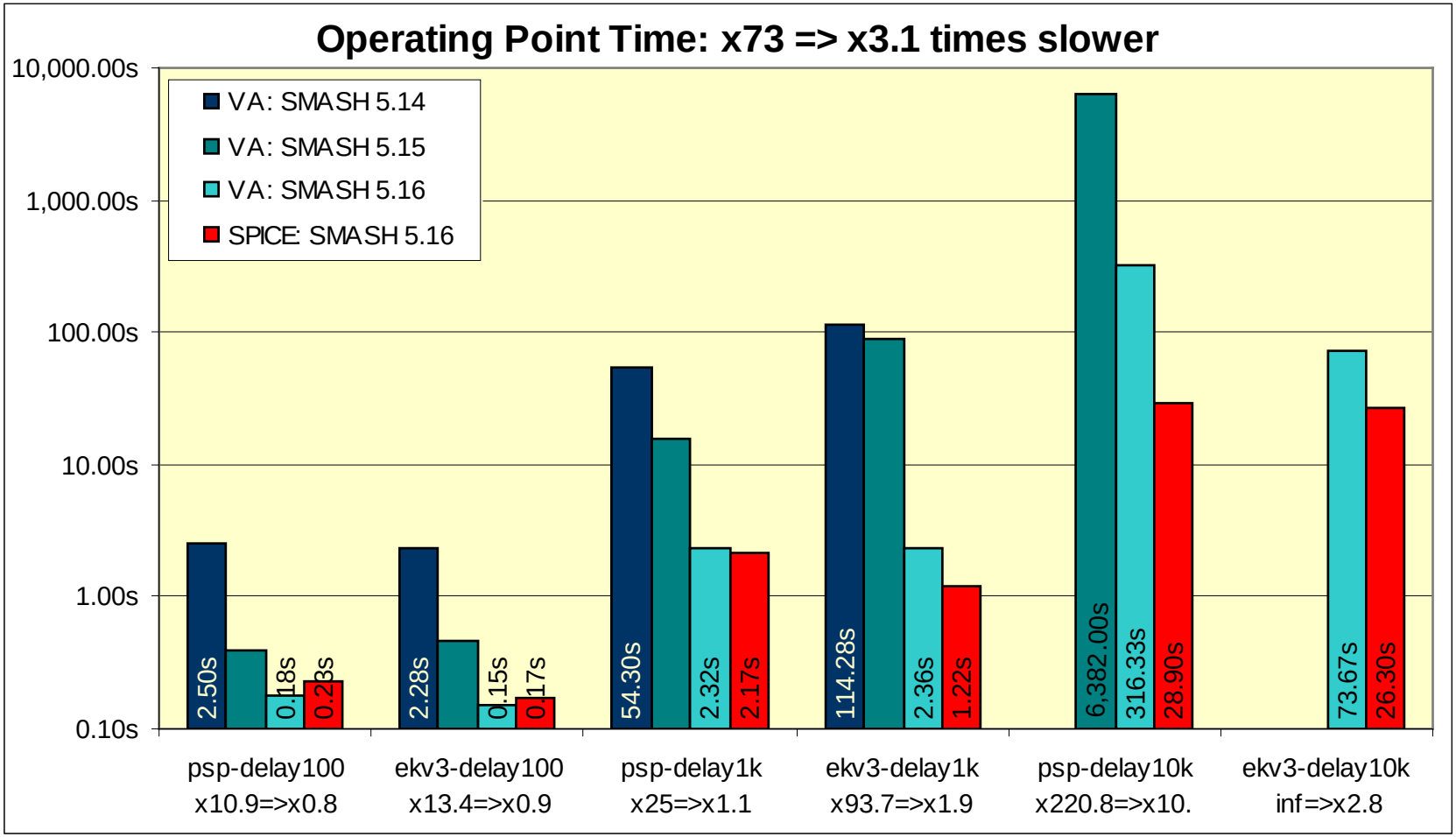
Test bench:

- Configurable CMOS delay (400, 4k or 40k MOS)
- Use default values for the parameters of the MOS models
- Use two models, one PMOS and one NMOS
- Computed iterations 2550±5
- Use TRAP method for integration

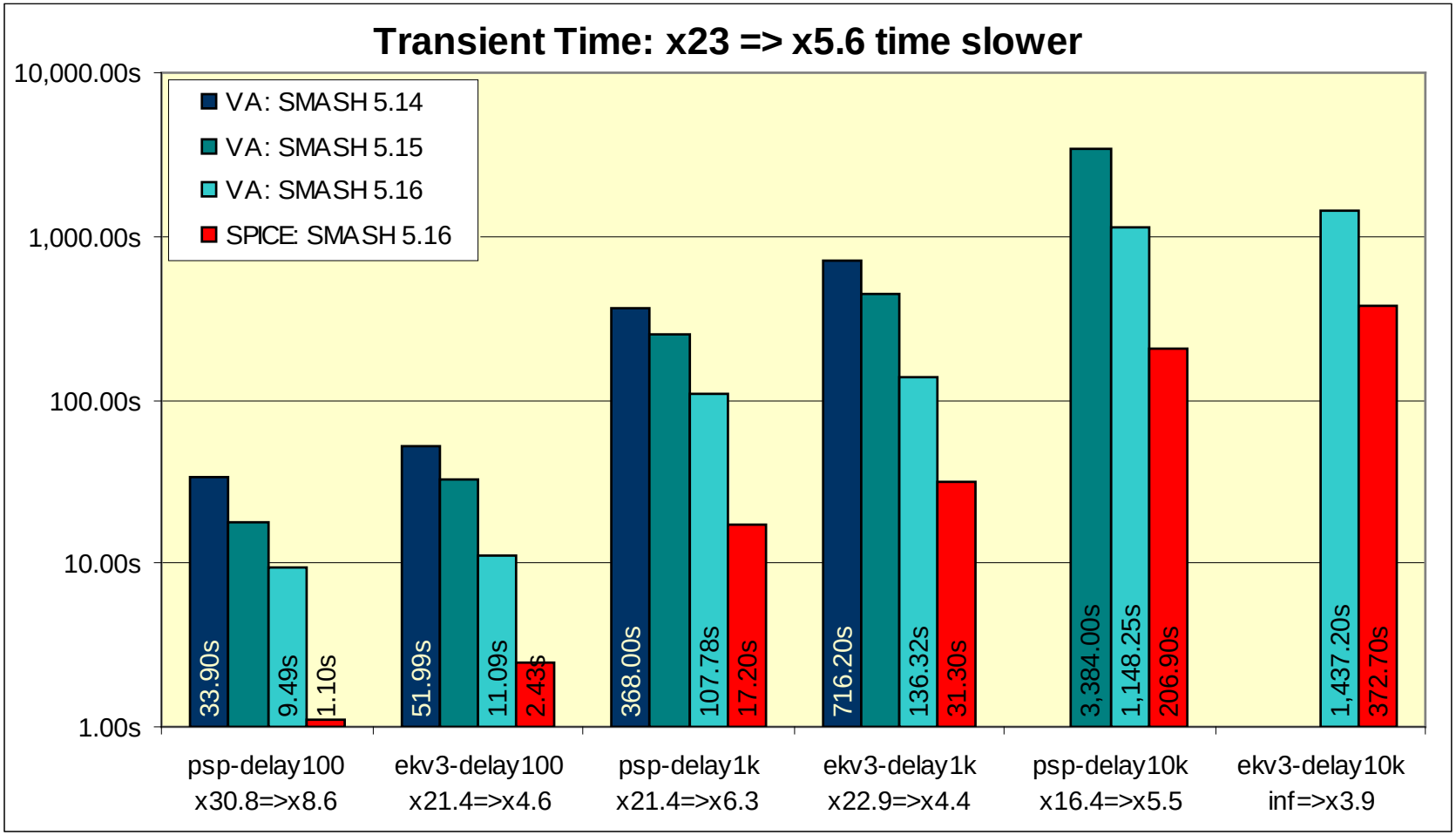
Progress in SMASH
Loading Time



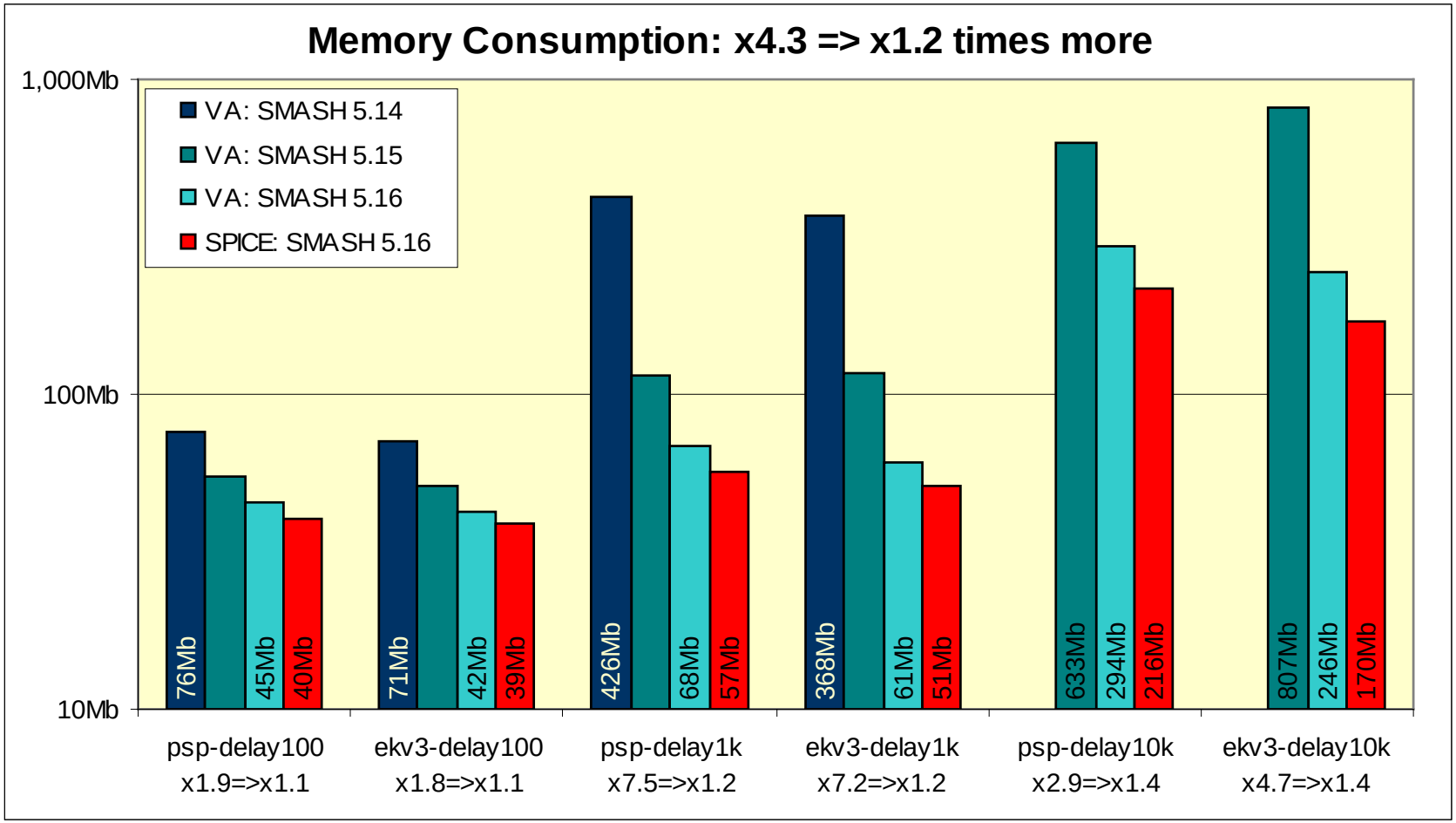
Progress in SMASH
Operating Point Time



Progress in SMASH
Transient Speed



Progress in SMASH
Memory Consumption



Benchmark of Verilog-A vs. SPICE Summary

- Loading Time
  - SPICE remains 5.5 times faster
  - Goal: catch up with SPICE in the two years to come
- Operating-point Time
  - SPICE remains 3.1 times faster
  - Goal: catch up with SPICE in the year to come
- Transient Speed
  - SPICE remains 5.6 times faster
  - Goal: catch up with SPICE in the six months to come
- Memory Consumption
  - SPICE consumes 20 % less memory
  - Goal: catch up with SPICE in the year to come

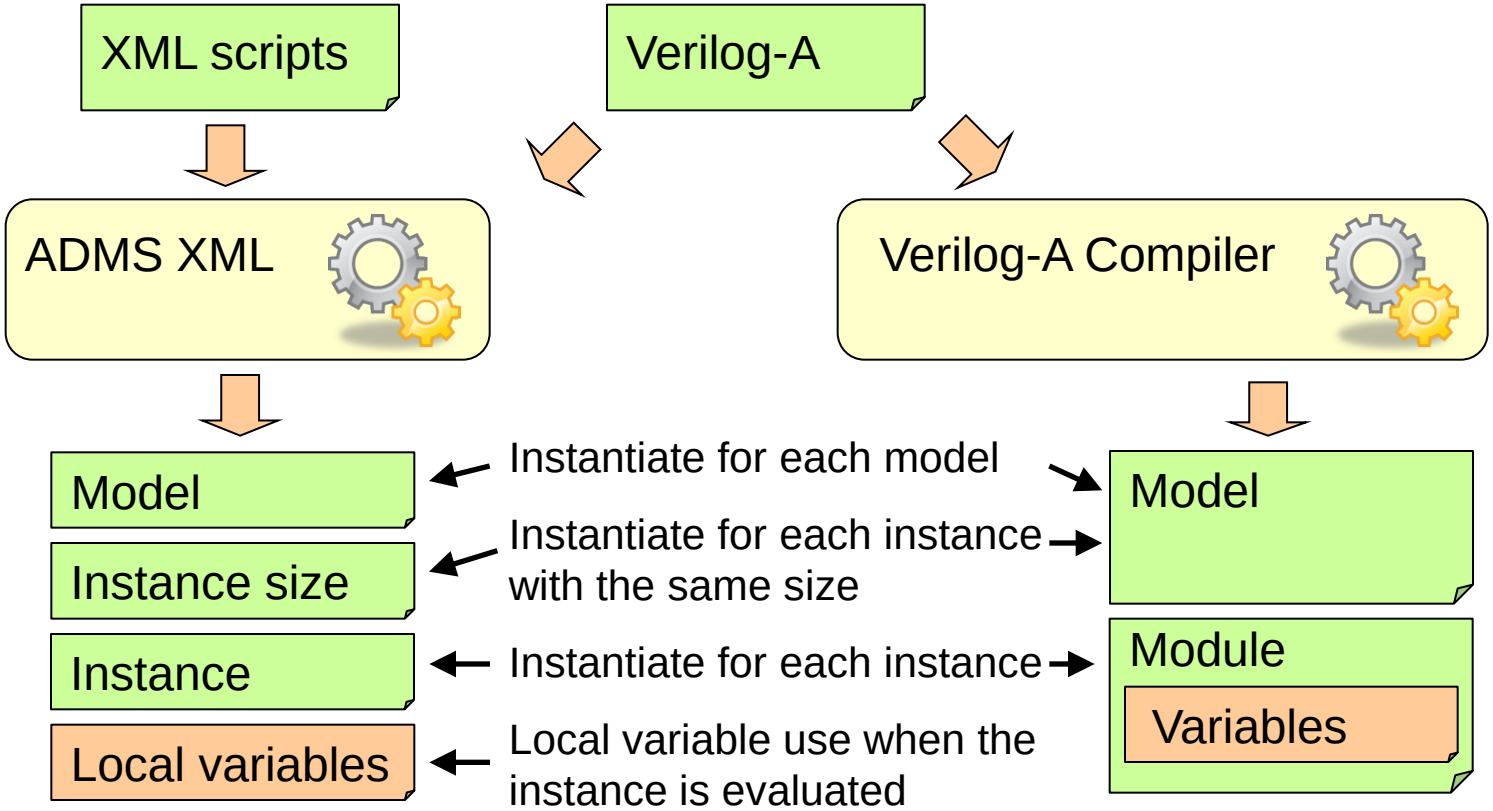
General purpose compiler vs. integrated devices

Verilog-A Limitations Simulation Speed

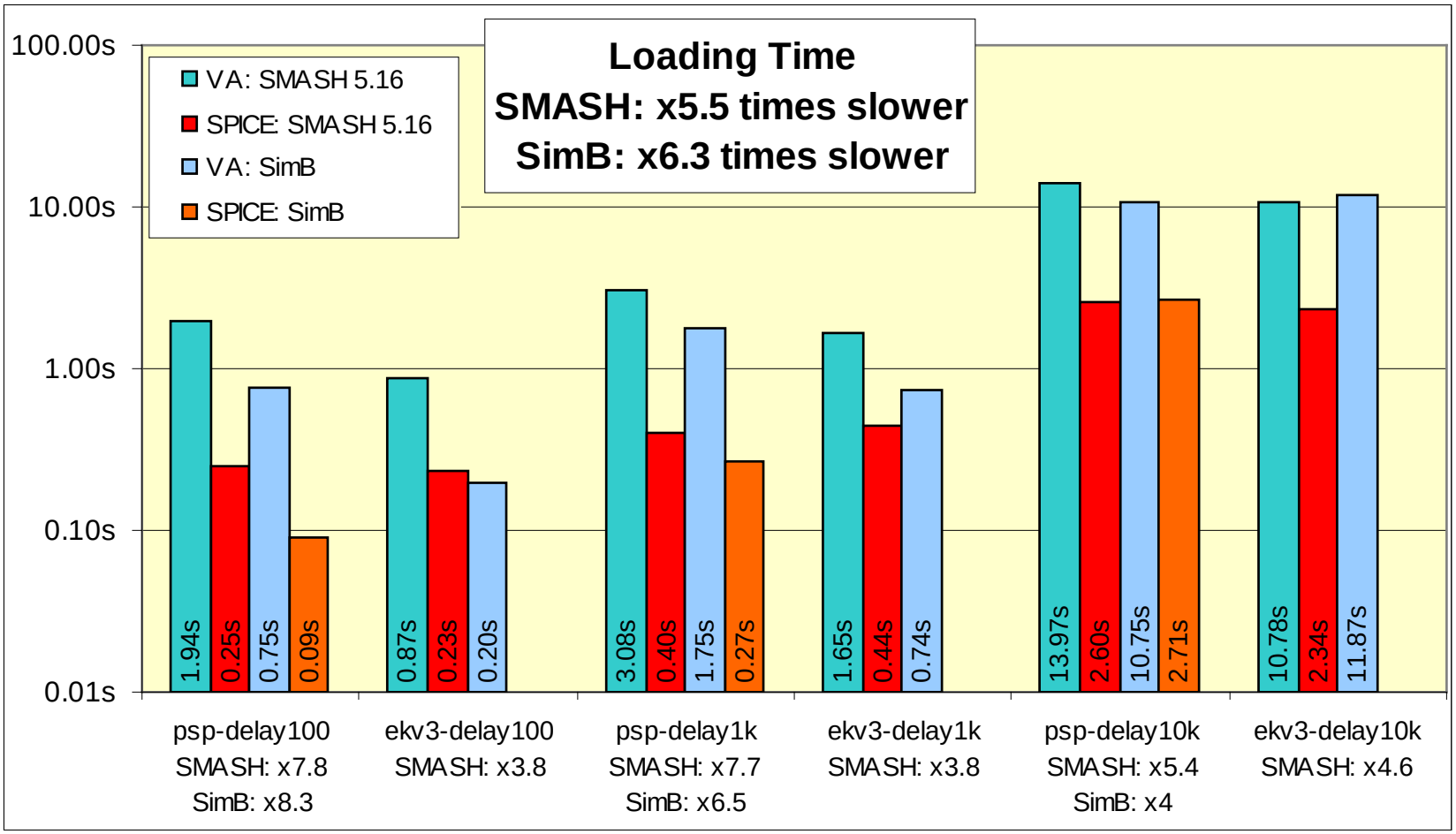
- Implementation dependent
  - Bypass/Linearization
  - Iteration specific code vs. specific code (initialization, noise)
  - Extra nodes
    - added for correlated noise due to ADMS XML limitation
  - Flow/Potential branches
  - Derivation/Integration
  - Hidden states
- Language (or coding) standard dependent
  - Iteration specific code vs. specific code (output variables)
  - Conditional nodes (collapsible nodes)

Verilog-A vs. SPICE

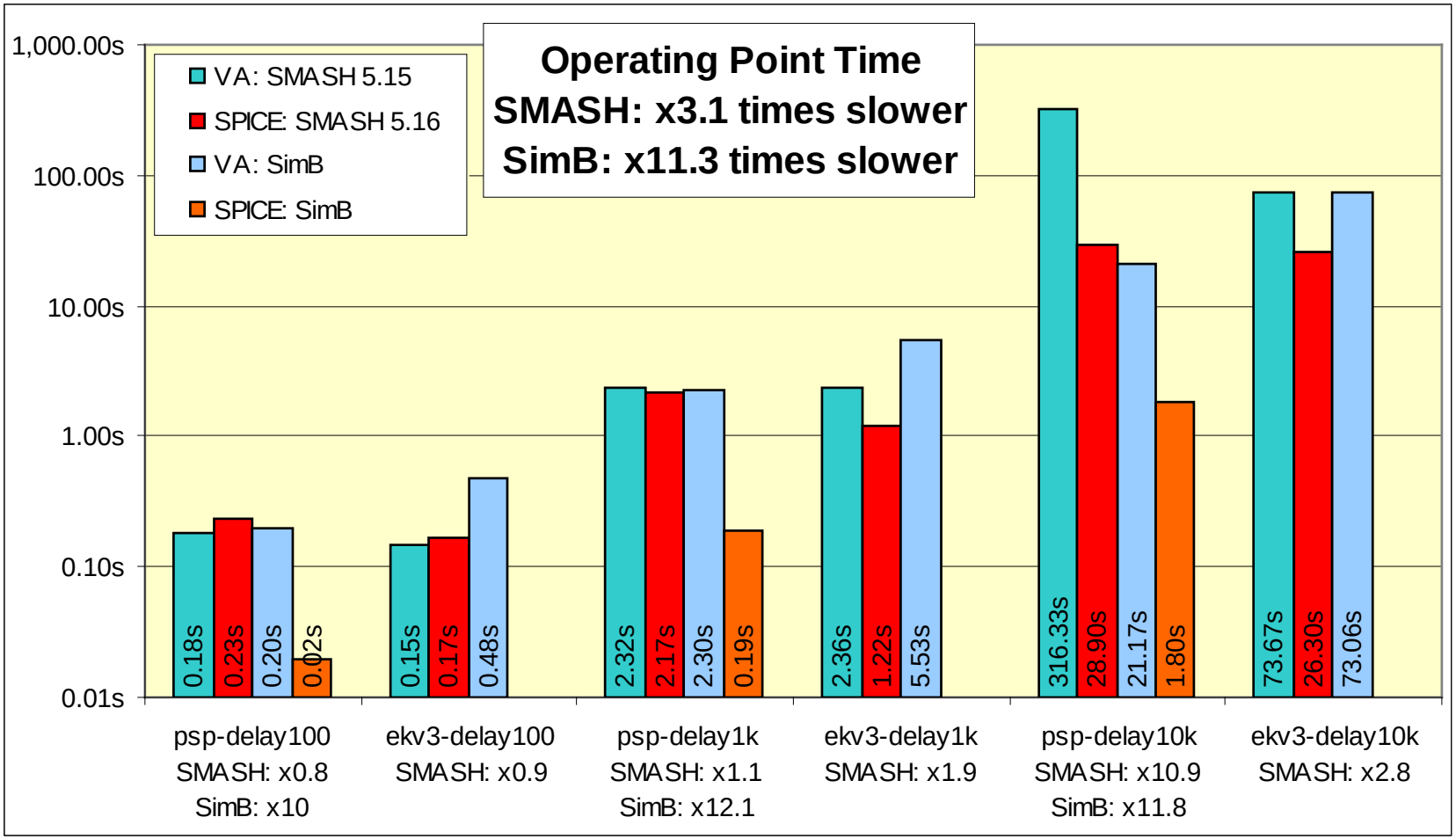
- Both start from Verilog-A description



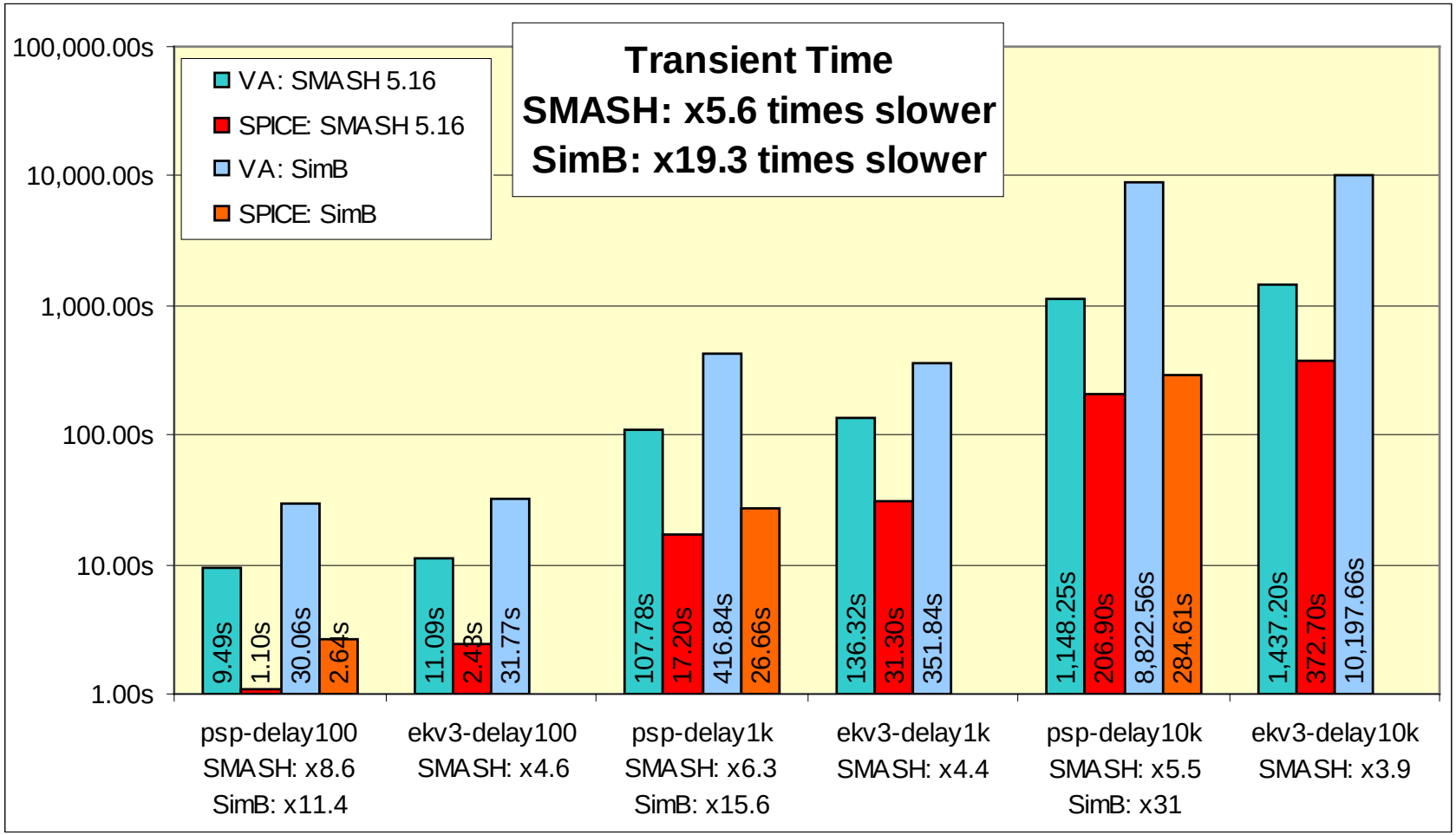
Benchmark of Verilog-A vs. SPICE Loading Time



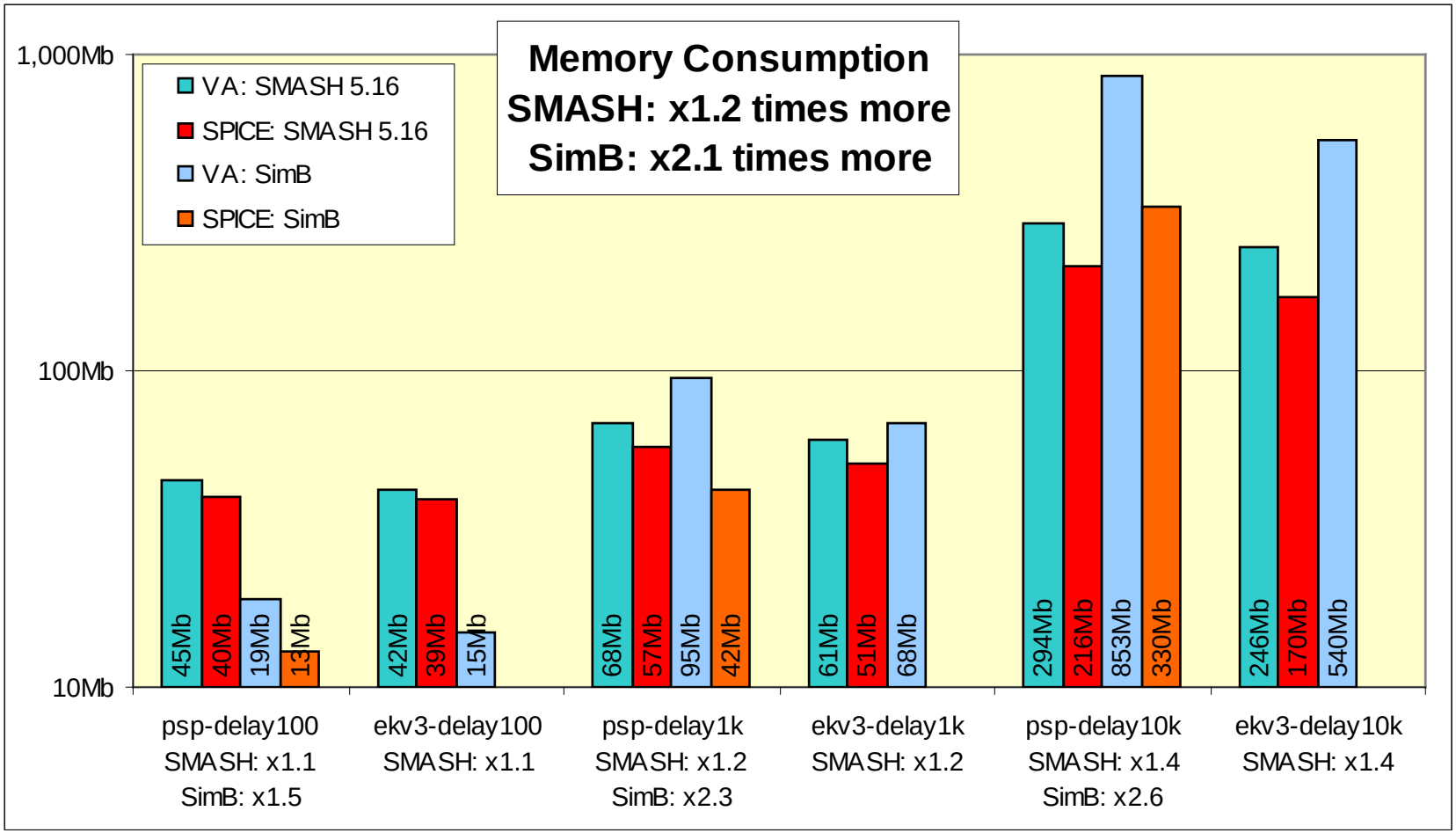
Benchmark of Verilog-A vs. SPICE Operating-Point



Benchmark of Verilog-A vs. SPICE Transient Speed



Benchmark of Verilog-A vs. SPICE Memory



Conclusion
Status

- For the moment, SPICE simulators remain faster than their Verilog-A counterparts
  - Compact models in Verilog-A should continue to target SPICE simulators and respect the inherent constraints to facilitate their integration into different SPICE simulators.

- What we have today
  - Verilog-A effectively adopted by Compact Model developers
  - Very efficient integration of CM into SPICE simulators using dedicated (ADMS-XML) or optimized Verilog-A compilers
  - A comprehensible approach for behavioral modeling of analog designs (same concepts in Verilog-A as in SPICE)
  - Very flexible mixing of SPICE with Verilog-A (as per LRM Annex E) allowing progressive behavioral modeling

Conclusion
Perspectives

- EDA vendors are filling the gap between SPICE and Verilog-A simulators
  - Progress during the past year in SMASH gives good perspectives for the future
  - EDA vendors must now make Verilog-A more attractive than SPICE for semiconductor foundries as well as for final users

- Future work
  - Integrate coding checks into general purpose Verilog-A compiler
    - Provide analog behavioral model developer with on-line feedback to help debug and optimize models
  - Integrate compact model optimization mode into general purpose Verilog-A compiler
    - Enable direct use in SPICE simulator of fully optimized compact model described in Verilog-A (no EDA vendor integration phase needed)

Conclusion
Outlook

- Verilog-A is in competition with non-public “API” approaches
  - To address the problems of deep submicron processes such as dynamic degradation, power consumption, system-level complexity…
  - Semiconductor foundry models currently developed as wrappers around compact models instead of extensions (in Verilog-A)
    - SPICE sub-circuit wrappers, TMI approach…

- Verilog-A has the potential to revolutionize the paradigm of analog design of integrated circuits and totally replace SPICE
  - Depends on the adoption of Verilog-A by all concerned actors: EDA vendors, compact model developers, semiconductor foundries as well as final users
  - HOWEVER, the compact model optimization mode is declared in the Verilog-A LRM but not defined

Please go ahead, download and use the free
"SMASH Discovery"
option from the DOLPHIN web site

Thanks!

Please send feedback about the
"Verilog Compact Model Coding Guidelines"
to DOLPHIN

Dolphin Integration
Meylan - France
Dolphin Integration GmbH
Duisburg - Germany

DOLPHIN INTEGRATION

www.dolphin-integration.com
medal@dolphin-integration.com