

# **Portable High-Performance Analog Models Using Verilog-A**

**Boris Troyanovsky, Patrick O'Halloran, and Marek Mierzwinski**

**IEEE Conference On Microwave Theory and Techniques,  
Philadelphia, PA, Jun. 2003**

Tiburon Design Automation, Inc.  
[www.tiburon-da.com](http://www.tiburon-da.com)

# Outline

---

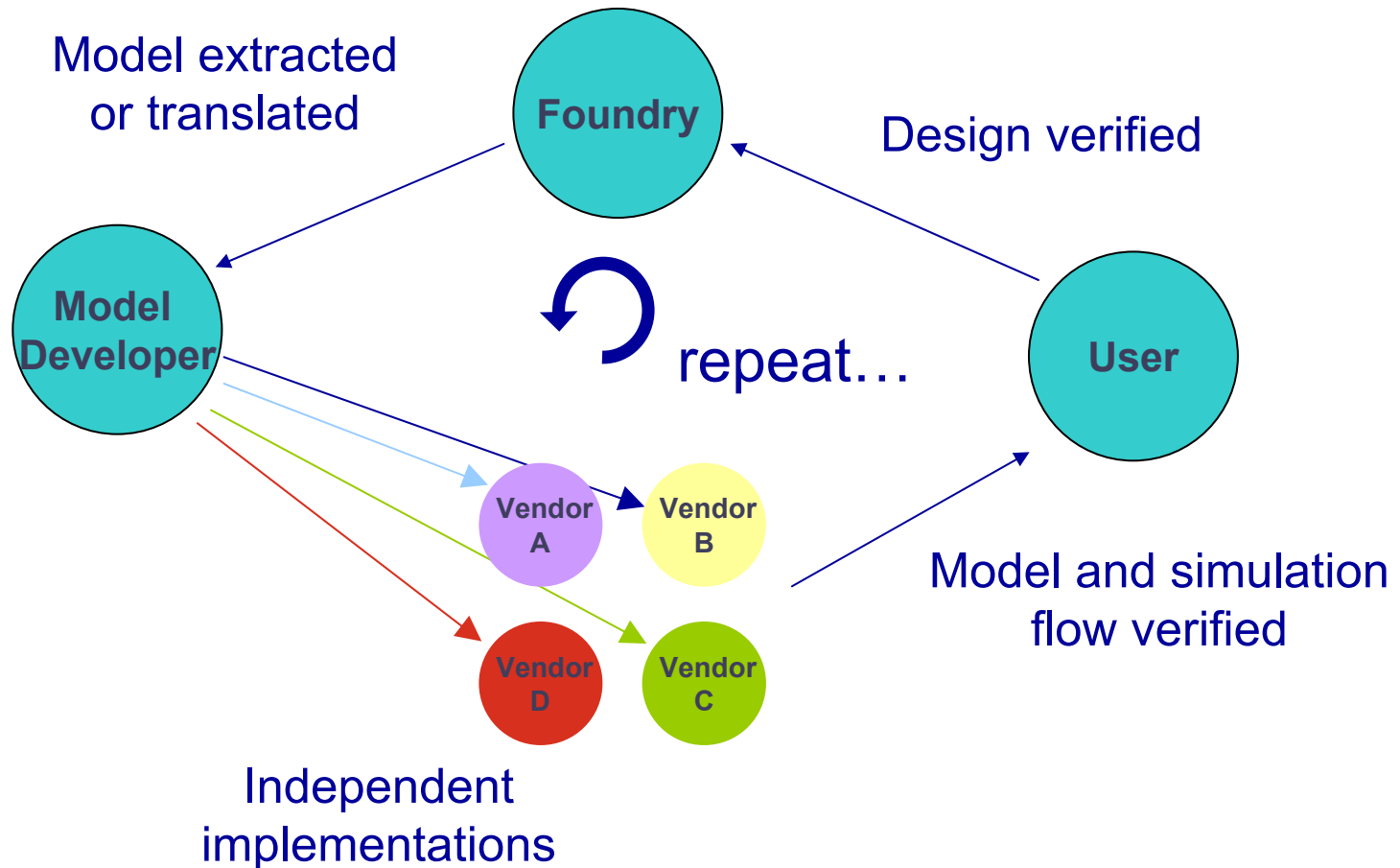
- Current state of analog model development
- Analog HDLs -- a comprehensive solution
- The Verilog-A language: a powerful, portable, flexible vehicle for analog modeling
- Examples
- Extensions and future directions
- Summary and conclusions

# Analog Model Implementation Today...

---

- High-performance / high-complexity semiconductor device models are implemented as “built-ins”
  - Interface to the simulator is proprietary
    - 3rd-party model developers may not have full access
    - implementations commonly done by vendor working off of source code for different simulator
  - Different simulators may have very different interfaces
  - Burden on model developer to
    - hand-calculate derivatives
    - be knowledgeable about numerous specialized analysis types
    - handle all the “software engineering” details and overhead (sweeping, optimization,etc.)

# Compact Transistor Models: Typical Flow



# Analog Behavioral Models

---

- Behavioral-level model libraries are widely available
  - Models include mixers, amplifiers, flip-flops, phase detectors, etc.
  - Some circuit simulation tool providers supply generic model libraries written in analog HDLs
  - Other providers supply models in proprietary formats and internal languages
    - often using proprietary capabilities and extensions, especially in the RF and microwave area
- Component-specific (nonlinear) behavioral models are not widely distributed
  - In the high-frequency and microwave area, component models are usually limited to relatively simple amplifiers and mixers

# The Consequences

---

- Long release times for models
- Subtle (and not so subtle!) differences between nominally identical models from different vendors
- A small number of standard models
- Behavioral modeling is not as far along as it could be

# Analog HDLs -- A Solution

---

- Languages are geared towards the model developer
  - Offer portability across simulator platforms
  - Shield model developer from the idiosyncrasies of each simulator model interface and each analysis type
  - Automatically handle error-prone tasks such as derivative computation
  - Allow the model developer to focus on model development rather than on software engineering
  - Facilitate easy transition between the various levels of modeling abstraction (behavioral vs. transistor-level)
- Currently, two leading candidates -- Verilog-A(MS) and VHDL-AMS

# Keys to Wide-Scale Adoption

---

- Performance
  - The speed of execution for compact transistor models must be comparable to that achieved by existing model interfaces
- Standard conformance
- Availability across the full spectrum of simulators and simulator analysis types
- Development environment
  - A good Verilog-A specific editor is very useful
  - A flexible debugger is highly desirable

# The Verilog-A Language

---

- Natural language for model development
- Concise
- OVI standard (proposed to IEEE)
- Implemented in many simulators
- Appears to be the leading analog-HDL candidate

# A Verilog-A code Snippet

---

```
module simple_diode(pos, neg);
inout pos, neg;
electrical pos, neg;

parameter real Area = 1.0 from (0:inf);
parameter real Is=1e-14, n = 2, Cjo=0;
parameter real Phi = 0.7, m = 0.5, tt = 1p;

real Id, Qd;

analog begin
    Id = Area*Is*(exp(V(pos, neg)/(n*$vt)) - 1);
    Qd = tt*Id + Area*V(pos, neg)*Cjo/pow((1-V(pos, neg)/Phi), m);

    I(pos, neg) <+ Id + ddt(Qd);
end

endmodule
```

# Some Verilog-A Features

---

- Analog functions
  - Help with modularity
- Behavioral modeling constructs
  - Event-triggered operators (crossing, timer)
  - delay, slew, transition, laplace, integration, etc.
- Support for hierarchy
- Support for non-electrical disciplines (thermal, mechanical, etc.)

# Existing Solutions

---

## Traditional Verilog-A solutions

- Are not sufficiently fast for simulation with typical compact models
- Are not universally available
- May not offer IP protection
- Don't support all analysis types

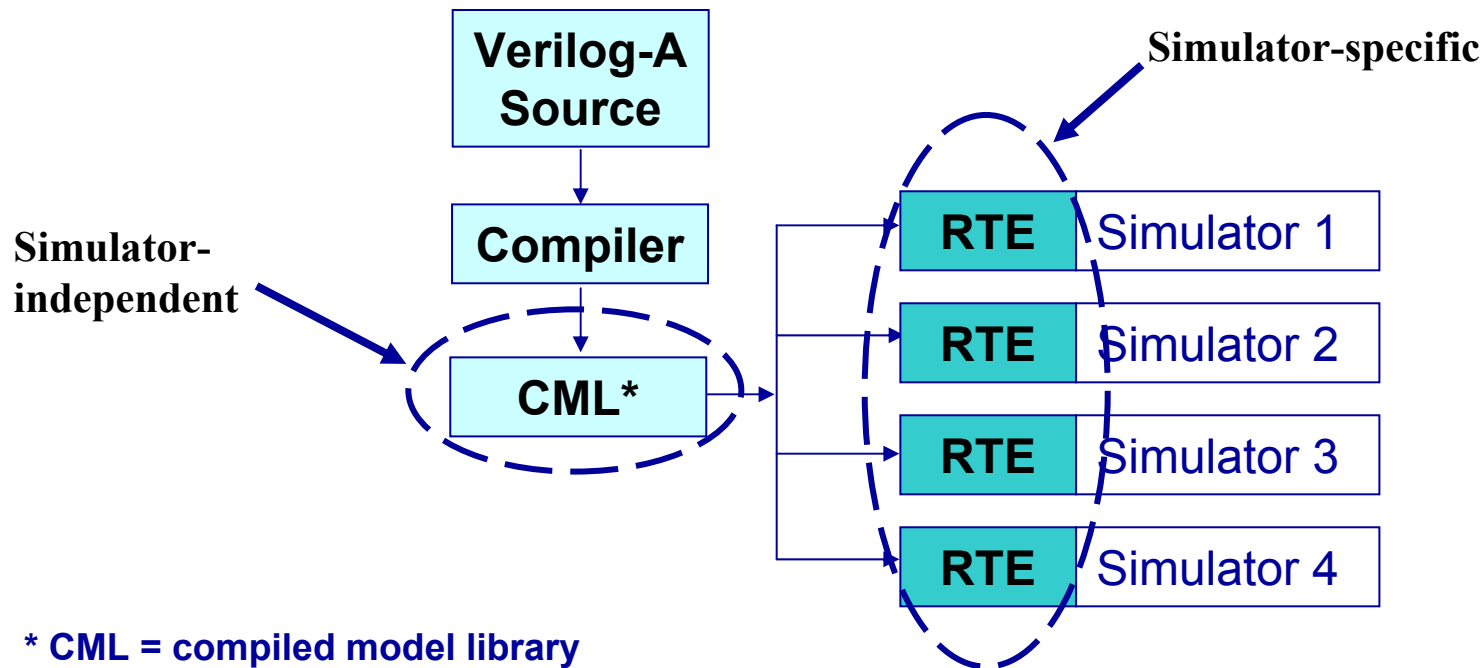
# Proposed Solution

---

- Use Verilog-A for analog model definition
  - Both compact transistor models and behavioral models
- Develop a Verilog-A model compiler and support tools which can be deployed across a wide range of simulation platforms
- Maintain simulation performance comparable to existing C/C++ level interfaces
- Support for all analysis types, e.g. transient, harmonic balance, shooting, nonlinear noise

# Proposed Architecture

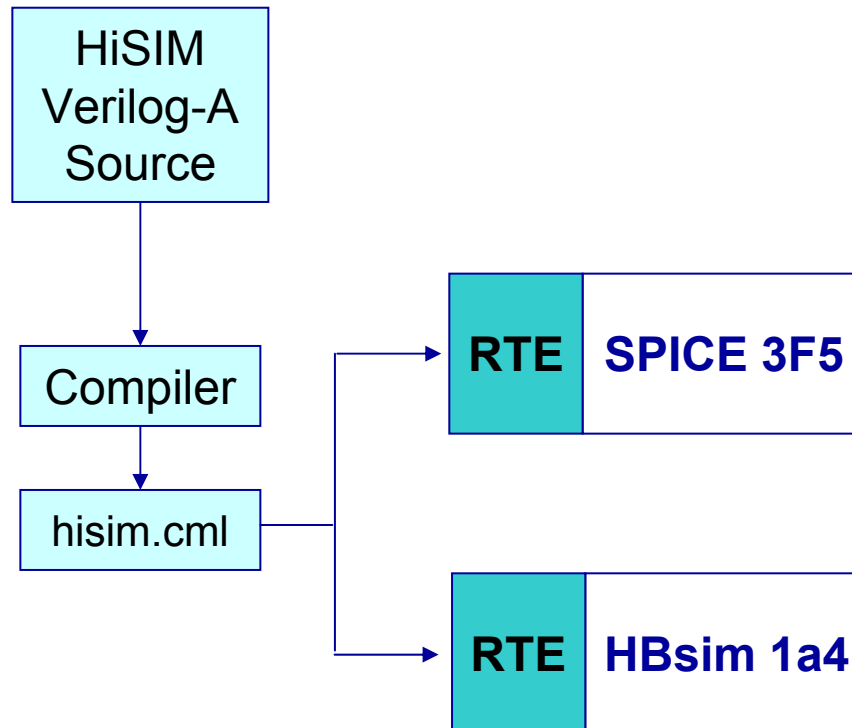
- A Verilog-A compiler and simulator-specific run-time environment (RTE)



# An Example: HiSIM Implementation

---

HiSIM: Hiroshima University  
STARC IGFET Model



# Benefits

---

- Benefits to end users
  - Solves the problem of model availability and compatibility
  - Models with source code available can be modified
- Benefits to vendors
  - Removes need to support complicated standard models.
  - IP protection for proprietary models available
- Benefits to model developers
  - Concentrate on model development, not implementation issues
  - Models and updates can be made available immediately on all platforms
  - Model IP can be protected and licensed

# Some Technical Implications...

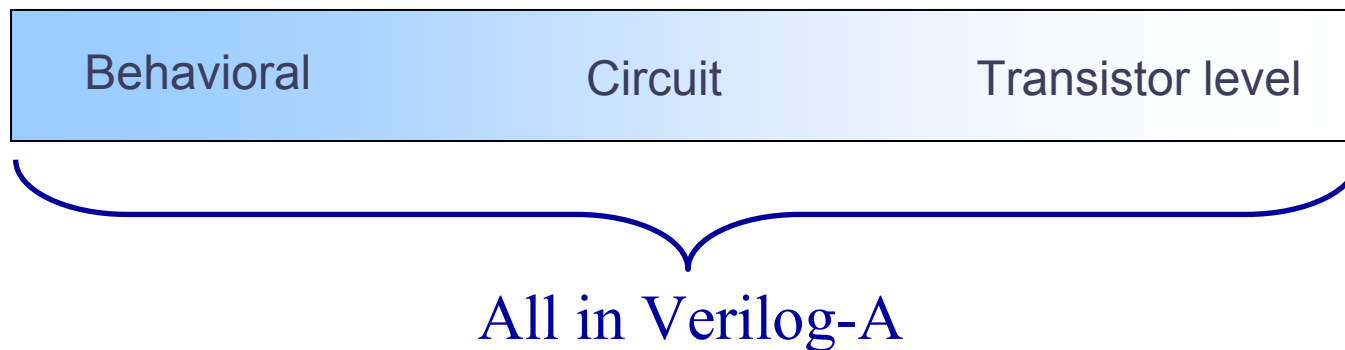
---

- Simulation vendors can focus more productively on algorithm innovation
  - New analysis types become easier to implement
- Accurate derivatives with respect to model parameters offer improvements in
  - Circuit optimization
  - Sensitivity analysis
  - Other similar areas

# New Design Capability

---

- Circuit designers can control level of abstraction
- All levels can be combined in one hierarchical implementation
- Designers can easily and transparently move between behavioral and transistor-level models of a component



# Compact Transistor Models Implemented

---

Implemented at Tiburon:

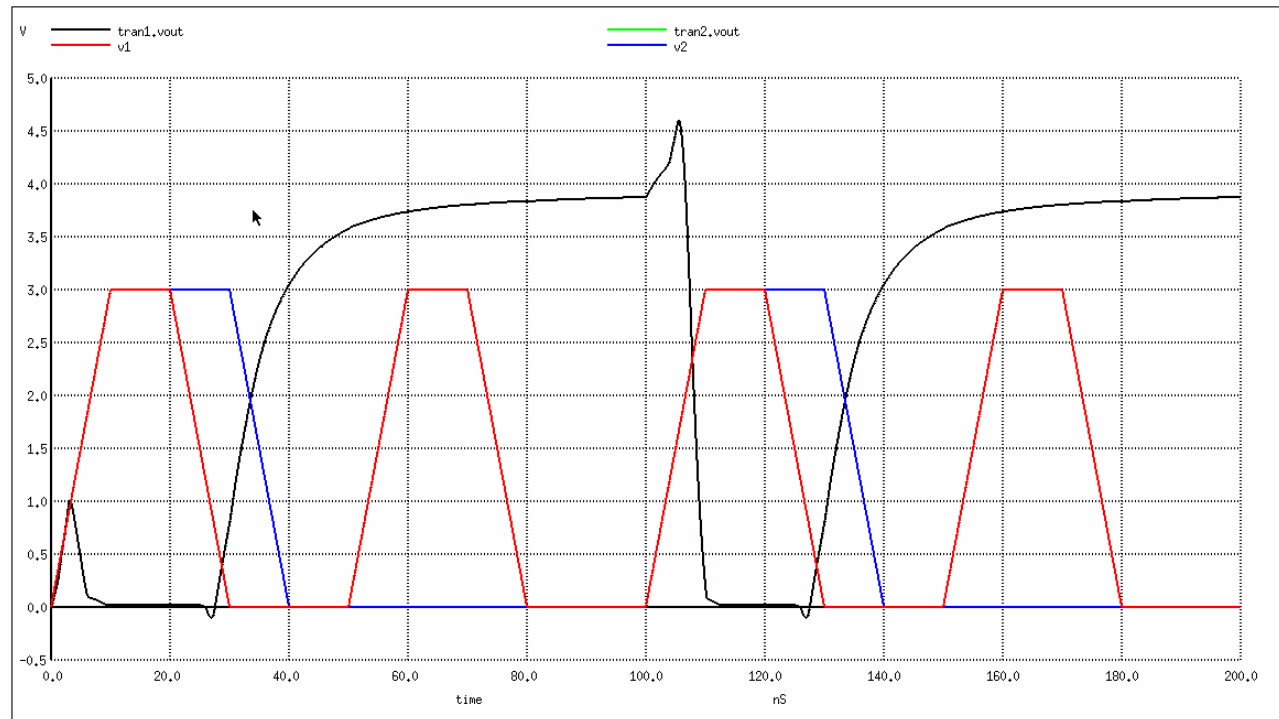
- BSIM3
- BSIM4
- SPICE Gummel-Poon
- Philips Mextram 504
- Philips MOS Model 9
- Philips MOS Model 11
- HiSIM

Available from others:

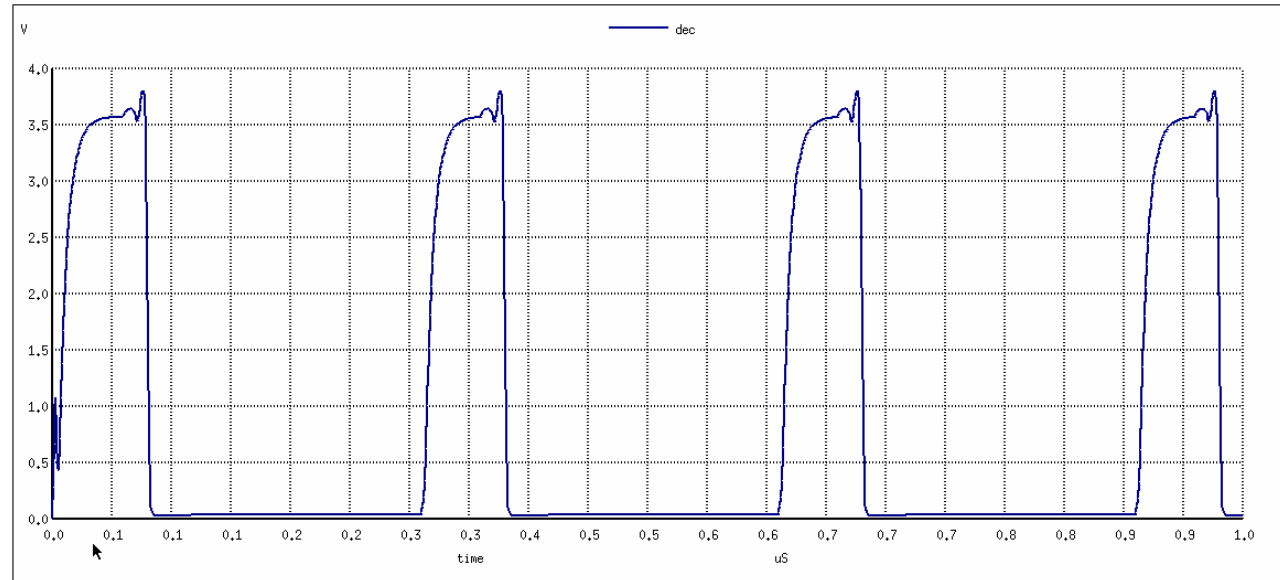
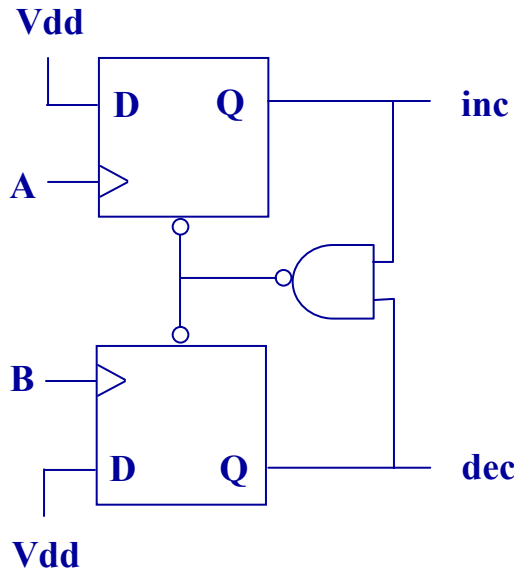
- VBIC
- EKV

# Example: Bipolar NAND gate (SPICE native BJT vs. Verilog-A BJT)

Standard  
bipolar TTL  
NAND gate



# Example: Phase-Frequency Detector



Circuit can be simulated at multiple levels:

Transistor level results shown above

Flip-flops and nand gate can be handled individually at behavioral level

The whole detector circuit can be handled at behavioral level

# Possible Future Extensions to Verilog-A

---

- Parameter specifications
  - Ability to specify parameters as being “required”
  - A convenient mechanism to indicate whether a parameter has been given
- Increased flexibility within analog functions
  - Support for pass-by-reference (in addition to pass-by-value)
  - Support for array-valued return types
- Optional ports (terminals) on modules
- Variable initialization

# Possible Extensions (cont.)

---

- Introduction of complex number type
- Support for generic convolution operator
  - Currently, language only supports rational polynomial approximations through usage of the `laplace_*` operators
- Other frequency domain extensions should be carefully considered
  - Efforts should be made to preserve the validity of constructs/features across all simulation analysis types
  - Ideally, all new constructs should be viable in both time- and frequency-domain analyses

# Conclusions

---

- Verilog-A has the capability to support complex analog model implementations
  - Both compact models and behavioral models can be efficiently supported
- A Verilog-A compiler provides fast execution and support across the full spectrum of analysis types
- Virtually all popular compact models have been implemented as a demonstration
- Model implementation and distribution can be greatly simplified
  - Model developers, simulation vendors, end users all benefit