The 3S Proposal: A SPICE Superset Specification for Behavioral Modeling

Michael Mirmak
Intel Corporation

June 5, 2007
Legal Disclaimer

THIS DOCUMENT AND RELATED MATERIALS AND INFORMATION ARE PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE. INTEL ASSUMES NO RESPONSIBILITY FOR ANY ERRORS CONTAINED IN THIS DOCUMENT AND HAS NO LIABILITIES OR OBLIGATIONS FOR ANY DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OF THIS DOCUMENT.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

Intel may make changes to specifications, product descriptions, dates and plans at any time, without notice.

Copyright © 2007, Intel Corporation. All rights reserved.
An Additional Disclaimer

• The following information is presented as the opinion of one person at Intel. This presentation does not necessarily represent Intel policy, commitments or preferences.

• This is not presented on behalf of the IBIS Open Forum and does not represent the official IBIS Open Forum direction.

*Other names and brands may be claimed as the property of others*
Agenda

• The What and Why of SPICE
• Analog Behavioral Modeling Today
• Pros/Cons of SPICE in General
• SPICE Compatibility
  – Elements as a Case Study
• Outline of a Behavioral SPICE Specification
  – What it must include
  – What it should exclude
• What a Standard SPICE Wouldn’t Address
  – Alternatives
• Summary

*Other names and brands may be claimed as the property of others
SPICE Review: A Tool and Modeling Method

- “Simulation Program with Integrated Circuit Emphasis”
- Developed by Donald Pederson at UC Berkeley in 1960s
- Not standardized, but the general format is widely recognized
- Berkeley still develops process models (BSIM3, BSIM4, etc.)
- SPICE 3F6 program available from Berkeley

- Many commercial SPICE flavors are available
- Most IC vendors have their own flavors, for their own processes
  - *Usually not compatible with any commercial SPICE variant*

*Other names and brands may be claimed as the property of others*
The Need for Analog Alternatives to “Old” IBIS

For a time, traditional IBIS (3.2/4.0) was “going out of style”

- IBIS is well-tuned to single-ended, simple designs (e.g., CMOS)
- IBIS 3.2/4.0 increasingly hard to use when modeling complex buffers
  - SerDes buffers with multi-tap equalization
  - Complex impedance modeling (frequency- and voltage-dependent C_comp)
- SPICE returned briefly as a popular alternative
  - IBIS Macromodeling Task Group emerged to support IBIS+ behavioral SPICE

*Other names and brands may be claimed as the property of others*
Why Use Behavioral SPICE for Signal Integrity?

- For buffers, address "traditional" (3.2/4.0) IBIS shortcomings above
  - IBIS 4.1/4.2 supports Berkeley SPICE code

\[ C_{\text{comp}} \neq \]

- Many vendors use SPICE to address IBIS package shortcomings
  - IBIS package support is clearly inadequate (see previous IBIS Summits)
  - ICM available but new; automated extraction for packages WIP
  - Integration into IBIS still not done

- IC/IP Vendor Needs
  - For complex systems, device/package models are not enough
  - Many customers expect full system decks, for analysis and correlation

*Other names and brands may be claimed as the property of others*
Which SPICE?

- In early 2007, IBIS-ATM took up the challenge
  - Asked a major proprietary SPICE vendor to release its manuals to the public
  - This was politely declined

Can we define a standard SPICE superset for behavioral modeling, “3S”?
"SPICE" Pros and Cons

- Something more than transistor models or traditional IBIS is needed
- How best to address advanced modeling and analysis needs?

For
- Can be used behaviorally
- Simple to understand
- Familiar to most engineers
- Versions implemented in most EDA tools
- Fairly flexible: if you can describe it with electrical elements, you can describe it with SPICE

Against
- Not standard
- A format, not a language
- Analog-only
- Not suited to algorithmic modeling (numeric processing instead of V, I analysis)
- Still data-driven

SPICE has a value for behavioral modeling. It is more flexible than table-driven IBIS with wider support at lower cost

*Other names and brands may be claimed as the property of others
Problems of Implementation: Elements

• Some Berkeley SPICE definitions are effectively universal...

<table>
<thead>
<tr>
<th>Element Prefix</th>
<th>Element Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-element</td>
<td>capacitor</td>
</tr>
<tr>
<td>E-element</td>
<td>VCVS</td>
</tr>
<tr>
<td>F-element</td>
<td>CCVS</td>
</tr>
<tr>
<td>G-element</td>
<td>VCCS</td>
</tr>
<tr>
<td>H-element</td>
<td>CCCS</td>
</tr>
<tr>
<td>K-element</td>
<td>mutual inductance/transformer</td>
</tr>
<tr>
<td>L-element</td>
<td>inductor</td>
</tr>
<tr>
<td>R-element</td>
<td>resistor</td>
</tr>
<tr>
<td>T-element</td>
<td>lossless transmission line</td>
</tr>
<tr>
<td>V-element</td>
<td>voltage source</td>
</tr>
<tr>
<td>X-element</td>
<td>subcircuit call</td>
</tr>
</tbody>
</table>

But evolution from Berkeley SPICE has caused deviations

• The elements below are completely different under different implementations

<table>
<thead>
<tr>
<th>Element Prefix</th>
<th>Conflicting Element Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-element</td>
<td>Non-linear dependent source IBIS element</td>
</tr>
<tr>
<td>O-element</td>
<td>Lossy transmission line Opamp</td>
</tr>
<tr>
<td>P-element</td>
<td>Semiconductor Resistor Port element</td>
</tr>
<tr>
<td>S-element</td>
<td>Switch element (voltage controlled) Multiport S-parameter t-line</td>
</tr>
<tr>
<td>W-element</td>
<td>Switch element (current controlled) RLG transmission line</td>
</tr>
</tbody>
</table>

• These elements are implemented in some tools but not all

<table>
<thead>
<tr>
<th>Element Prefix</th>
<th>Element Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-element</td>
<td>Current source</td>
</tr>
<tr>
<td>N-element</td>
<td>Lossy transmission line</td>
</tr>
<tr>
<td>U-element</td>
<td>Lumped lossy transmission line</td>
</tr>
<tr>
<td>Y-element</td>
<td>Macro element</td>
</tr>
<tr>
<td>Z-element</td>
<td>Frequency-dependent component</td>
</tr>
</tbody>
</table>
SPICE Problems

• Inconsistency
  – As shown above, even basic elements are inconsistent between SPICEs
  – Shared element functions are often inconsistent (e.g., V sources)
  – Some analysis functions are proprietary (e.g., S-parameter generation)

• Lack of expandability
  – Users cannot define new elements or analyses, only new subcircuits
    • At least one proprietary SPICE defines a “macro” element, but the macros are under the control of the tool vendor, not the user or author

• Lack of control
  – In an age of 10 Gbps signals, obtaining inconsistent results for “standard” models is no longer tolerable

These have been the hurdles to widespread SPICE usage for behavioral modeling
What Would A Standard Behavioral SPICE Look Like?

• Any standard SPICE would have to support the following:
  – “First-letter” name + node + function syntax for elements
    • Including the truly common elements and formats
  – Subcircuit syntax and approach
  – “Dot” syntax for functions, parameters and analysis types (e.g., “.OPTIONS”)
    • Common analysis types and functions TBD
  – Other common structural assumptions
    • No ordering requirements aside from title and .END

• What would be excluded from the standard definition?
  – Transistors and other active elements
    • Process files and “LEVEL” would also be excluded
    • Do we need diodes?
  – Any element name that is inconsistent among proprietary implementations
  – Functions or capabilities outside circuit solving
    • Field solvers, digital logic functions, links to other languages or tools

This is readily achievable
What Would A Standard Behavioral SPICE Add?

• Major additions
  – The A-element
    • Undefined in any SPICE, based on informal survey
    • Could be used as a specification-level “catch-all” macro alias
    • Allow 3S elements new to some SPICEs to be easily implemented through extended parsing in more sophisticated SPICEs
    • Specification would control associated function definitions

  \[ Axxxx \text{ macroname } n1 \ n2 \ \ldots \ \text{SPICE 1} \]
  \[ Nxxxx \ n1 \ n2 \ \ldots \ \text{SPICE 2} \]
  \[ Wxxxx \ n1 \ n2 \ \ldots \ \text{SPICE 3} \]
  \[ ?xxxx \ n1 \ n2 \ \ldots \ \text{SPICE 3} \]

  – .COMPAT/.UNCOMPAT switch
    • “Wrapper” for standard behavioral SPICE text
    • Netlist-level flag to tool to enforce behavioral SPICE standard rules
    • Enables use of 3S code within a proprietary netlist

These should not represent a significant industry burden

*Other names and brands may be claimed as the property of others
Issues

• A format-only specification may be insufficient
  – Different tools may still interpret data differently
  – Complex SI measurements (DDR2) would still be painful
  – IC/IP vendors want more control over analysis **methods** (algorithms)
    • *e.g.*, *causality and passivity enforcement on transmission lines*
  – This favors **language-based** rather than **format-based** approaches

• Can we truly exclude active devices (transistors)?
  – BSIMx is still highly popular, useful and effectively a standard

• Administrative burdens are considerable
  – Maintenance, including adding new macro functions, would be required
  – Development of a syntax parser would also be needed

Is this still worth doing?
Are there alternatives?

*Other names and brands may be claimed as the property of others*
“The Split”

- Both models and tools have to address varying market segments
- IBIS-ATM work shows that model creation and use are growing apart
  - System designers need relatively simple models in inexpensive tools
  - IC designers need more detail on the digital, data-processing side

**IC Vendors**

- Higher-cost, highly capable tools
- Digital and analog support needed
- Same environment for model development and testing
- Portability less important until export stage

**System Designers**

- Lower-cost tools
- Analog focus (no digital support needed)
- Model use, not development
- With multiple suppliers, model portability is key

*Other names and brands may be claimed as the property of others*
A Better Solution than SPICE?

- Verilog-A is already a SPICE superset
  - Verilog-A is already standard, widely supported (IBIS 4.2!) and available
    - At least one major SPICE vendor supports Verilog-A today
  - Supports analysis control without requiring digital language support
    - IC vendors get control over model interpretation
  - Addresses both the element naming and function definition problems
    - Element names are separate from instance names

```
IBIS_R #( .Rval(R_val), .Scale(Scale_val)) R1 (Node1, Node2);
```

Anyone recall the IBIS Macromodel library using Verilog-A for SPICE?

- Transmission lines and S-parameters are a significant omission
  - An enhanced Verilog-A could support SI features, system netlists

Is Verilog-A a more compelling analog modeling solution than SPICE?
If We Go Ahead with 3S, Next Steps...

1. Outline and document the basic features, common to most SPICEs
   - Element names and formats
   - Element functions
   - Options and analysis types
   - Other key syntax and netlist structural assumptions

2. Define the new features, unique to 3S
   - The A-element syntax
   - A-element macro names and data formats
   - .COMPAT/.UNCOMPAT usage

3. Define analysis requirements ("the hard part")

4. Create illustrative test cases

5. Write the relevant specification documents
   - Likely in parallel with the steps above

Delaying work on a unified analog solution means more "lost ground" for IBIS and standards

*Other names and brands may be claimed as the property of others*