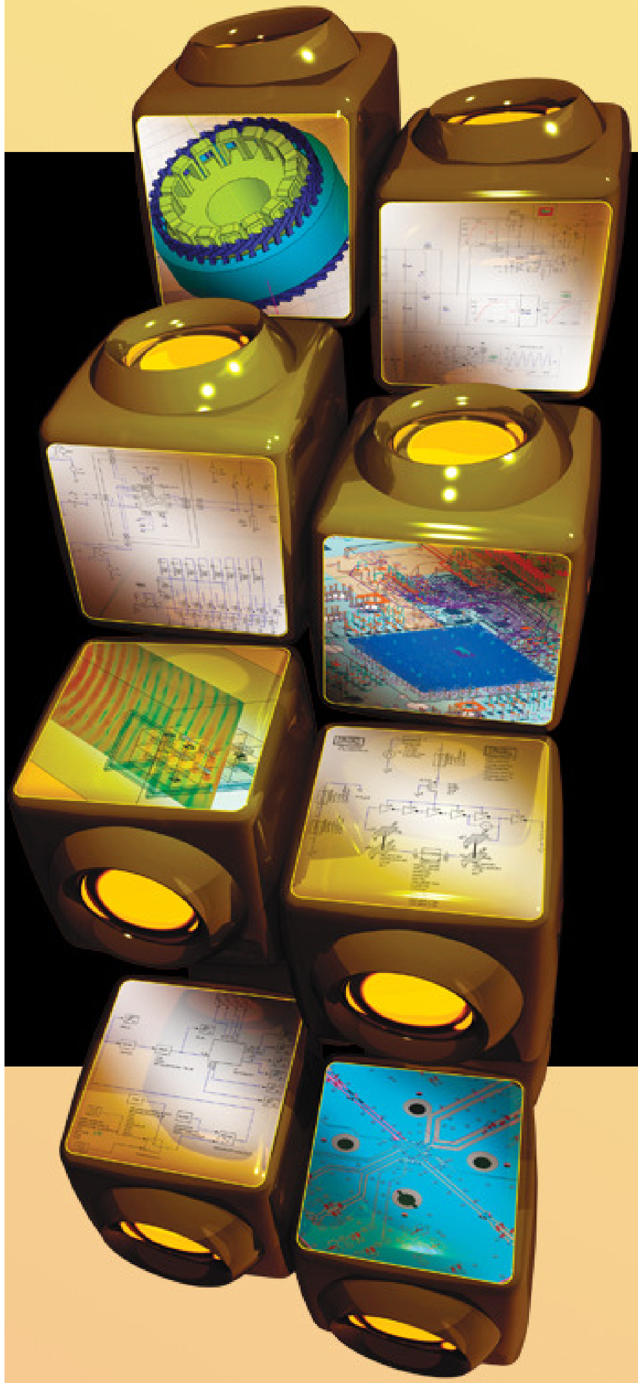

Inspiring Engineering

CMC Standard Spice Language and Model Format Committee

Samuel Mertens

Presentation to IBIS Committee

A subsidiary of ANSYS, Inc.

Compact Model Council

ANSOFT
A subsidiary of ANSYS, Inc.

- Charter: To promote the international, nonexclusive standardization of compact model formulations and the model interfaces.
- Have standardized BSIM3, BSIM4, BSIMSOI, HICUM, MEXTRAM, PSP, HiSIM_HV and soon HiSIM2 transistor models in conjunction with a university. Have also standardized a MOSVAR_CMC, Diode_CMC, R2_CMC and R3_CMC within subcommittee.
- 44 member companies in America, Asia and Europe
- 4 General Meetings, sub-committees have regular phone meetings
- Also working under GEIA
- Web-site: http://www.geia.org/index.asp?bid=597

Goal

- Create a standard language and model format with all existing functionality in a standardized formal way.
- There will be simulator lang=… directives to incorporate existing languages. Obviously, mixed cases will NOT work for all simulators.
- If one needs to go back to their previous language or model format because something is missing, we have failed

Standard Spice Language and Model Format currently represented in subcommittee

- ADI
- Agilent
- Ansoft
- Atmel
- Cypress
- Freescale
- IBM
- LSI
- Maxim
- Mentor Graphics
- National Semiconductor
- NXP
- Panasonic
- Sony
- Toshiba
- Synopsys

Inspiring Engineering

Standard Spice Language and Model Format Status

- We have put together a list of requirements. This is included at end of presentation.
- We hope this will help us select a basis for the standard spice language and the model format.
- Right now we have 2 candidates as a basis for this language:
  - Spectre
  - Verilog A

Standard Spice Language and Model Format Requirement Highlights

ANSOFT
A subsidiary of ANSYS, Inc.

- The language must be defined by a formal grammar.
- The language must be case-sensitive.
- The language will define basic components needed in design kits.
- The language must not contain positional arguments (except the terminal list may be position dependent).
- A key letter must not be used to identify a component type. Full names (not level numbers) must be used.
- The model file format should have similar functionality as the Verilog A paramset semantics.
- The language will have built-in support for seamless integration of multiple technology files
- The language must support the notion of hierarchy and scope.

Inspiring Engineering

Standard Spice Language and Model Format Schedule

A subsidiary of ANSYS, Inc.

Proposed Schedule:

- Finalize requirement list by Q1-09: done
- Choose basis for language by Q2-09
  - Using the requirement list we will select how the language should look.
  - Need to complete examples in different formats so we can make a choice
- Release standard language by Q4-09

Standard Spice Language and Model Format Question to IBIS Committee

- Are you interested in working together on this? How can we help each other?
- What are your requirements?
- Adding support for IBIS models? Do you have a proposed format?
- Adding S- or W-element support? Do you have a proposed/preferred format?

For more information

E-mail Samuel Mertens: smertens@ansoft.com

Requirements: General

1. The language MUST be defined by a formal grammar.
2. The language MUST have unambiguous definitions of a model card, an instance of a model, a subcircuit, and a library.
3. The language MUST be case-sensitive. The standard SHOULD specify a warning mode, which would warn users when names are being redefined which only vary in their capitalization.
4. The language MUST define the standard floating point format that is to be used.
5. The language MUST understand units prefix symbols (defined in Appendix 1). Unit names and symbols MUST NOT be allowed to be a part of the instance/model card.

Requirements: General (continued)

A subsidiary of ANSYS, Inc.

6. The language SHOULD be able to use international characters sets. The language MUST define the allowed ASCII characters that may be used in names, and provide an escaping mechanism for specifying other characters.
7. A mechanism to define global nodes MUST be part of the language. The language SHOULD specify how these nodes are defined within a hierarchical structure.
8. The language MUST define legal characters and name spaces, including any length limitations. Reserved keywords that can't be used within their namespace for variable/instance/parameter/node names must be defined.
9. The characters that start a comment, both for a full line and for the remainder of the line MUST be defined in addition to the characters that are used to continue multi-line statements, white space characters, tab spacings , embedded comments, single line comments, block comments and extra spacing. The language SHOULD allow a minimum of such characters.

Inspiring Engineering

Requirements: General (continued)

10. Basic models (R, L, C, mutual inductors, independent sources and controlled sources) and Verilog-A, API generated, or other built-in models MUST be instantiated in a standardized way (Appendix 2).
11. The language MUST NOT contain positional arguments (except the terminal list may be position dependent).
12. The terminal list for a component MUST have a default order. This order MUST be able to be overridden by a specific connectivity directive (recommendation: use the Verilog standard). A warning/parsing mechanism has to be defined if the user does not connect the required nodes.
13. The terminal list for a component MUST (optionally) be defined by delimiters (recommendation: use parentheses "()"). The committee will decide if this is optional or not after exploring examples.

Requirements: General (continued)

14. A key letter MUST NOT be used to identify a component type. Full names (not level numbers) must be used (Appendix 2). If a model uses a parameter named LEVEL to distinguish between versions or model features, it is to be treated as a model parameter, not to distinguish it from another model. If a new version of the model is different enough from a previous model, the model developer will have the option to give it a different name to identify it.
15. The language MUST be able to provide one model card as a subset of another one, including remapping of the model parameters to new names.
16. The model file format SHOULD have similar functionality as the Verilog A paramset semantics - a lot of our technical requirements are actually already satisfied by this format, such as chaining, automatic model selection (by overloading).

Requirements: General (continued)

A subsidiary of ANSYS, Inc.

17. The language SHOULD allow conditional instantiation, also with scoping.
18. The model file format SHOULD allow multiple model sets to be used next to each other, for example allow the support of BICMOS and carrier laminate technologies in one simulation environment. Different technology files SHOULD have a different namespace. The language SHOULD have a mechanism to allow the user to switch between these namespaces when generating a netlist. The language needs to clarify what happens when two model sets both define a variable called “foobar” at the top level, or both want to set a global option or if more than one technology file is loaded which contains a model with an identical name.
19. The language MUST allow a model to be defined within a subcircuit, and SHOULD allow to reference this model from another subcircuit. The language SHOULD NOT allow a subcircuit to be defined within another subcircuit, unless more evidence and examples can demonstrate the need for this.

Inspiring Engineering

Requirements: Limitation for first phase

20. The standard will NOT define analysis specific parameters for sources.

Requirements: Parameters, functions, referencing, hierarchy and scoping

21. The language MUST support the notion of hierarchy.
22. The language MUST support the notion of scope. It MUST appropriately propagate variables defined at a higher hierarchical level to lower hierarchical levels ("global" variables MUST be included in this). This is required for multi-technology simulation.
23. The language MUST define netlist parameters, netlist functions and expressions. It MUST define the allowable data types and the allowed operators (+ - * /,…) and functions (log10,ln, sin, …) MUST be defined.
24. The language MUST define the syntax and semantics of to refer to instances, models, parameters, nodes and branches, including references at any level of hierarchy, in and out of the subcircuit definition. The language MUST support a method to alias part of the hierarchy, similar to the inline functionality.

Requirements: Parameters, functions, referencing, hierarchy and scoping (cont.)

25. The language MUST define how to pass currents and voltages to all instances and subcircuits where this is needed (eg Verilog A). The currents could be defined by a probe or a source.
26. The language MUST be able to handle both global statistical variables, common across all components affected by a specific process parameter, and mismatch statistical variables that affect individual instances of components.
27. The language MUST allow instance parameters to be evaluated inside of models. Especially, the multiplicity factor MUST be explicitly available, it is required for proper specification of mismatch variation, and MUST be handled hierarchically. To clarify, the multiplicity factor MUST be accessible and usable in expressions, where the simulator properly expands any hierarchical definitions of multiplicity. For models where the implementation of the multiplicity factor has caused confusion, the standard will provide a specification.
28. The language MUST be able to define statistical relations between parameters in specific statistical parameter sets. The language MUST NOT define how the analysis is implemented.

Requirements: Interaction with simulator/environment

ANSOFT
A subsidiary of ANSYS, Inc.

29. The language MUST contain syntax to specify a limited set of simulator options, for example “tnom”,” temperature” or“scale.” These MUST be able to be specified hierarchically. The language SHOULD specify how these simulator options can be set or accessed within different technology files. The set MUST be defined within the standard (Appendix 3). Where possible the name of such parameter SHOULD correspond with its Verilog A equivalent.
30. Encryption directives SHOULD be defined.
31. The language SHOULD have a structure that allows easy integration in source code documentation tools such as Doxygen.

Inspiring Engineering

Requirements: Interaction with simulator/environment (cont.)

ANSOFT
A subsidiary of ANSYS, Inc.

32. The language SHOULD allow for a preprocessor to handle macros and conditionals. This would mean we have to define preprocessor functionality. The language SHOULD not include scripting, although a mechanism MAY be defined for access to names and values supplied from an exterior shell.
33. The language SHOULD allow parameters that exist only to help convergence or other simulator-specific behavior to be labeled as such so a simulator would be able to recognize, or ignore, them.
34. The language SHOULD be identifiable as such, but not by a particular file name extension. This could be achieved through simulator lang=cmc_standard directives.
35. The language SHOULD support the ability to do range checking of parameters and bias dependent parameters, and issue warnings and errors.