Panel: The Next-generation HDL

Submitted and Organized By:
Richard Goering
EE Times
Felton, California 95018
(408) 335-3390, (408) 335-3579 — FAX rgoering@cmp.com

Nanette Collins
Marketing and Public Relations Counsel
Boston, Massachusetts 02115
(617) 437-1822, (617) 425-0340 — FAX nanette@nvc.com

The recent market battle between VHDL and Verilog HDL has obscured the larger issue of whether either of these languages is adequate for next-generation design. Can they be extended, or do we need an entirely new language that’s constructed to handle system-level design descriptions, behavioral and logic synthesis, simulation, and formal verification? If so, how should that language be created?

Moderator:
Steven E. Schulz, P.E.
Senior Member, Technical Staff
EDA Systems Group
Texas Instruments
6146 Prestonshire Lane
Dallas, Texas 75225-1911
(972) 917-3648
ses@dadd.ti.com

Panelists:
Kurt Keutzer
Chief Technical Officer
Synopsys Inc.
700 East Middlefield Road
Mountain View, California 94043-4033
(415) 694-4356
keutzer@synopsys.com

Position Statement: The Next Generation HDL

The next generation HDL cannot simply describe hardware; it must describe system functionality which comprehends both hardware and software. It seems fair to say that in most electronic systems built today the effort in the software content far outweighs the effort in the hardware content. Just as dog collars are created to fit the dog and not its tail, it is likely that the system-design language of the near future will cater to needs of the software developer and not to idiosyncrasies of the hardware designer. Finally, to meet the immediate for a software-oriented system-design language, system developers are turning to existing programming languages and their software-development environments. As a result, in the near future, systems will be described in standard programming languages such as C++ or perhaps Java. The real challenge for CAD tool developers is not to idly speculate about a future over which they have little control, but to get down to the real work of creating synthesis and simulation environments to accommodate a software-dominated world.

James A. Rowson
Fellow
Alta Group of Cadence
555 N. Mathilda Avenue
Sunnyvale California 94086
408-523-4157, 408-523-4601 — FAX jimmr@altagroup.com
Position Statement: Interfaces and Refinement are Key to new Design Language

The explosion of silicon capability combined with shrinking design cycles requires the industry to find a way to make reuse viable. With our current design languages, it is difficult, if not impossible, to build Intellectual Property (IP) that is reusable at the system level because communication ports on the IP must be described by their implementation (by listing their pins).

A new description method is needed to help separate the currently intertwined notion of behavior and communication. The communication with a piece of IP should be described in terms of transactions, similar to the level provided by a device driver. The implementation of that transaction can be handled in many ways using many different encodings in time and space. By raising the abstraction level of communication, we can make all MPEG decoders, for instance, look identical at the system functionality level. They will vary only in the way the implementation delivers information to the decoder.

By separating communication from behavior, we can provide a methodology that will make possible incremental refinement from high levels of abstraction down to a concrete controllable implementation. System-level verification can be done using transaction-based models that will run extremely fast, being unburdened by details of implementation. As concrete IP is selected, the specific glue logic and state machines necessary to glue the IP together is incrementally defined and verified. Without a new description language, and the separation of communication from behavior, this incremental refinement will not be possible.

Larry Saunders representing VHDL International
Principal
SEVA Technologies
9340 Carmel Mountain Road, Suite D
San Diego, California 92129
(619)538-6283, (619)538-4271 — FAX lfs@seva.com

Position Statement: Do we really need a “Next Generation” HDL?

VHDL and Verilog HDL have come to dominate hardware design over the past 10 years. Now some are contending that it is time to define a new “system description language.” While it’s easy to say “out with the old and in with the new,” shouldn’t we first consider whether HDLs we already have are being used effectively? And, what about repercussions of developing yet another HDL, when billions of dollars worth of existing industry infrastructure are at stake?

Most hardware designers are just getting comfortable with existing HDLs and HDL-based tools/methodologies. What new HDL capabilities will offer a compelling reason to switch: behavioral modeling, system-level constructs? VHDL and Verilog HDL already do that. The truth is that existing tools and methodologies don’t even begin to tap into the descriptive power available in VHDL and Verilog HDL. Creating yet another HDL that isn’t effectively
utilized by existing tools and methodologies isn’t going to help anybody.

Alberto Sangiovanni-Vincentelli
Department of EECS
University of California at Berkeley
Berkeley, California 94720
(510) 642-4882, (510) 643 5052 — FAX
alberto@eecs.berkeley.edu
or at:
Parades
Via di San Pantaleo 66
00186 Roma
Italy
+39-6-68807923, +39-6-68807926 — FAX
alberto@leonardo.parades.rm.cnr.it

Position Statement: Basis for a New Design Language
As the electronic design landscape changes to include multi-million gate chips built using embedded processors and other complex reusable IP, it is becoming more and more clear that current description languages need to be overhauled. Any new language must be implementation neutral, tool neutral and model of computation neutral.

Implementation neutrality is important because all new interesting systems being defined include a substantial software component. Any new design language needs to support equally well components built in hardware or software. Many product derivates are distinguished by what software items are moved into hardware (and vice versa) for cost, power or business reasons. Without a neutral description, this migration cannot be done by reusing components — they have to be redone from scratch.

Tool neutrality is necessary because designers cannot afford to build multiple models for any reason, let alone because tools need subtly different versions of the same model. Synthesis, simulation and formal verification all need to be supported equally well, none should be the sole driver.

At high levels of abstraction, multiple focused models of computation are required for specialized problems to be found there. Examples include dataflow and reactive systems. Any new design language needs to be able to mix and match different models of computation within a single system.

Maq Mannan
Chairman, Open Verilog International
Director, Design Automation
National Semiconductor Corp.
2900 Semiconductor Drive, MS D3677
Santa Clara, California 95052
(408)721-6340, (408)773-0978 — FAX
mannan@galaxy.nsc.com

Position Statement: Evolution of a New Design Language
By definition, a Hardware Description Language needs to define and describe hardware with all its constraints. The next generation of this “HDL” will come from the evolution of Verilog HDL because of its flexibility in handling “software”-type descriptions, its strength in all areas of design hierarchy and its worldwide acceptance as the premier HDL of today.
Position Statement: Basis for a New Design Language

So far, high-level languages designed for hardware generation seem to have not fully benefited from the progress in programming language semantics on which several modern software languages are based. We shall argue that theoretical and practical tools for mathematical semantics are mature and that HDL design should be based on them. We shall discuss our view of the synchronous computation model and of its use in synchronous languages design (Esterel, Lustre, etc.).