

One Language or More?

- How Can We Design an SoC at a System Level -

**Panel Organized
by
Masaharu Imai
Department of Informatics
and
Mathematical Sciences
Osaka University**

Panel moderated
by
Gary Smith
Chief Analyst
Dataquest
(Member of the SLDL Working Group)

The emerging Core Based Design Methodology is forcing engineers up into the Electronic System level of design. It is only there that they can make the hardware and software trade-offs needed for optimal design performance.

Unfortunately, as we move up into ES Level Design, our two hardware description languages, VHDL and Verilog, are running out of steam. The design language dilemma is, "Which language will fill our need for a new ESL language?" Some start-ups are pushing alternate hardware languages, while others are approaching the problem with modifications of software languages. The modifications are necessary, as software languages are sequential languages. A hardware design language must be concurrent.

This panel will discuss the merits of various languages and various methodologies being presented to solve this problem.

Panelist:

Steven Schulz – Texas Instruments (SLDL)
Karen Bartleson – Synopsys (SystemC)
Daniel D. Gajski – UC Irvine (SpecC)
Wolfgang Rosenstiel – University of Tuebingen (Java)
Peter Flake – Co-Design Automation (Superlog)
Hiroto Yasuura – Kyushu University (C, C++)
Masaharu Imai – Osaka University (VHDL)

Points of Discussion:

- How can we model, refine, and optimize as SoC design at the Electronic System level ?
- Does any one language, such as Java or C++, have an ability to perform this job ?
- Or do we need a set of languages to do this job, such as proposed in SLDL ?
- What are the requirements for the SoC design language(s) ?
- How well can we model, refine, and optimize the design using the(se) language(s) ?

