The Role of Verification at the Architecture Definition Stage

Tuesday, September 20 | 3:45 pm – 5:00 pm
Panelists

- **Sean Smith**, Chief Verification Architect, Denali Software
- **Brian Bailey**, Independent Functional Verification Consultant, Brian Bailey Consulting
- **Debashis Chowdhury**, VP-Research & Development, Synopsys
- **Amir Hekmatpour**, Senior Engineer, IBM Microelectronics
- **John Pierce**, Senior Architect, Cadence Design Systems
- **Paul Tobin**, Senior Manager, AMD
What should you verify?

• Verification is NOT ensuring that the final design conforms 100% to the understanding of the RTL designer (basically designer’s interpretation of the architecture).

• Neither is verification done to ensure that the RTL code has no software bugs.

• Verification is establishing that the design (implementation) conforms 100% to the architecture definition (intent).
Why wait, then, till the RTL is ready?

• The verification engineer should master the architecture and not the RTL code.

• In the architecture definition phase, the design can be modeled at a much higher level of abstraction – easier to write tests patterns which can run later, on the RTL. The high level models, developed at the architecture definition stage, can serve as golden models and accelerate randomization & self checking tests.

• Performance bottlenecks can be detected at the architecture definition stage when there is much more flexibility to modify the architecture.

• Verification engineers can even request for architecture changes that will help them verify the design more easily.
The earlier verification starts, the better it is

• Since the verification folks never know when to stop, let us at least know when to begin!

• The earlier the better

  ▪ Let us not get biased towards the RTL code

  ▪ Let us not ask the RTL guys for a list of bugs they are expecting in their code and then write test patterns to see if they got it right (what if they miss a few bugs – will you also?)

  ▪ What if the architecture has bugs!