System Level Design Tools:
Who needs them, who has them and how much should they cost?

Brian Bailey
Chief Technologist
Design Verification and Test Division
Why go through the pain of abstraction shifts?
Where is the bottleneck today?

- Verification is the point of most pain (4.5/5.0)
  - It's also the weakest link in the methodology (62%, 71% of failures)
  - First time functional failures increased from 30+% to 62%

- In a recent survey, where engineers were asked for their top issues:

  Functional verification time 59.43%
  Quality and reliability 58%
  Time-to-Market 47.49%
  System integration time 27.92%

(multiple choices were allowed, so figures do not add up to 100%)
Executable Specifications

- Design and Verification require 2 independent descriptions of the system.
Direction for the future

Declarative Description of Intent

Executable Specification

Synthesis

Design Flow

Predictor of a Verification Flow

Test case generator

Functional Coverage

Auto Checker

Properties for Formal V
Back to the Questions
System Level Design Tools:

- Who needs them?
  - Very few people on the creation side.
  - In the verification path would get used by everyone

- Who has them?
  - Today, for data path and algorithm design
  - A few for architectural analysis
  - Not really possible for verification today

- How much should they cost?
  - How much value do they provide?
    - If it affects 10% of the process, then not much.
    - If it affects the 70% of time spent on verification, then a lot more.