

# VMMing a SystemVerilog Testbench by Example

**Ben Cohen**
**Srinivasan Venkataramanan**
**Ajeetha Kumari**

**http://www.abv-sva.org/**
**ben@abv-sva.org**

# Objectives (This presentation)

Ben Cohen
abv-sva.org

- **Goals**
  - Experiences in creating a VMM compliant testbench
  - Demonstrate key applications of VMM features
- **Roadmap**
  - DUT is FIFO *(in SV with SVA)*
  - SystemVerilog Testbench with VMM for
    - Generation of random transactions
      With transactor, through channel
    - Consumption of transactions from channels
      With transactor
    - Creation of monitor transactions
      Through channel, for consumption by scoreboard
- **Not Addressed here**
  - Factories
  - Callbacks

# Language Supported by a Methodology

Ben Cohen
abv-sva.org

- **RVM is proven methodology based on Vera**
  *(RVM is Synopsys Reference Verification Methodology based on Vera)*
  - **Framework for testbench software**
  - **Transaction-based & coverage-based**
  - **Supported by reusable library**
- **VMM is adaptation of RVM for SystemVerilog**
  - **Transaction-based & coverage-based**
  - **Supported by reusable library**
  - **Documented in books**
    - ***Verification Methodology Manual for SystemVerilog, 2005 Springer***
    - ***SystemVerilog For Verification, A Guide to Learning the Testbench Language Features,*** Chris Spear, 2006 Springer

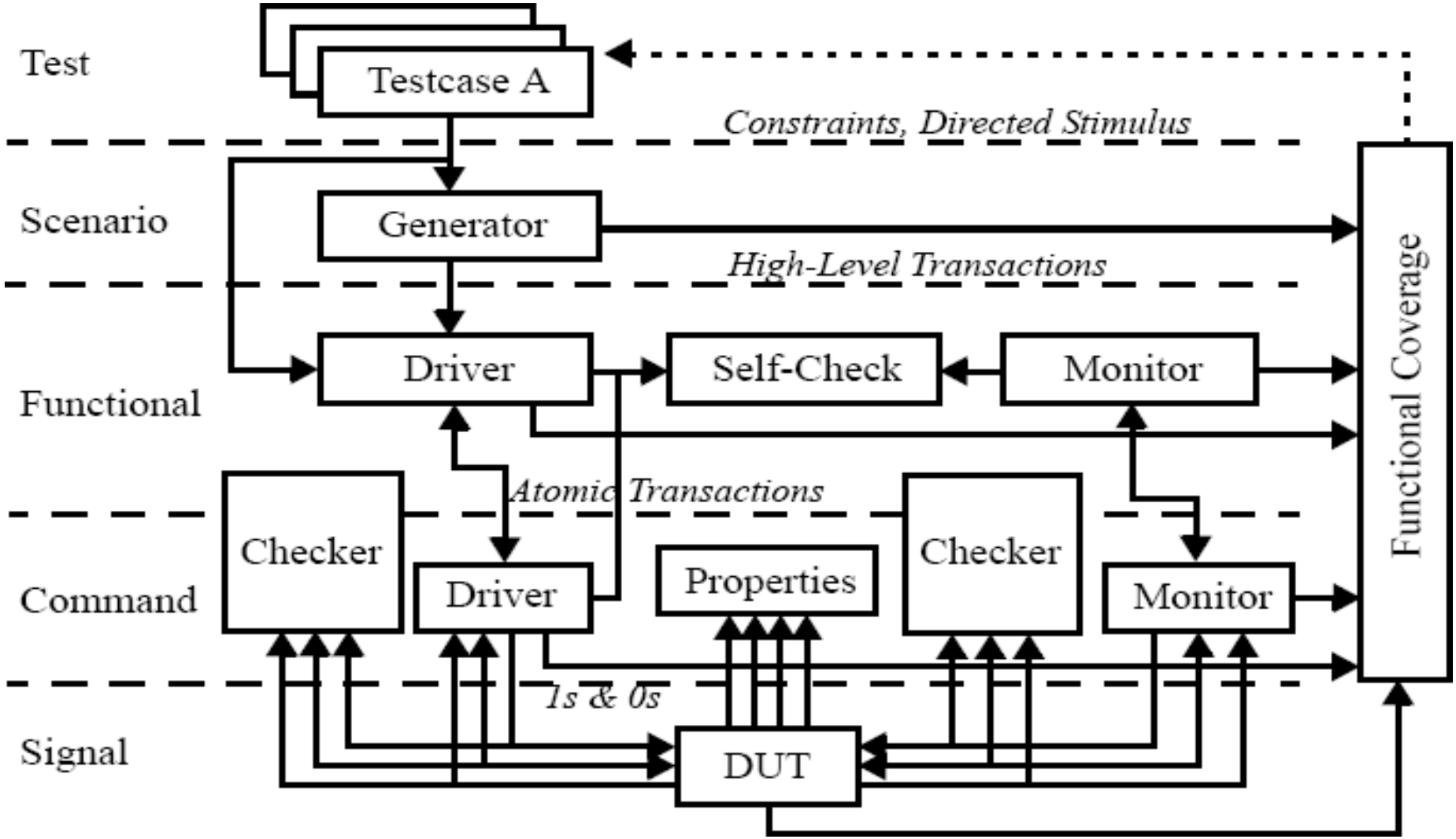
# Goals of VMM

Ben Cohen
abv-sva.org

- **Verification productivity**
  - **Framework supports a consistent testbench flow**
    - **Structural aspect**
      - transaction / channel / transactor / environment
      - Monitor / scoreboard
      - Testbench construction
      - Automation in construction (macros)
    - **Run flow**
      - Execution sequence
      - Scenarios
      - Random and directed tests

# Layered Verification Environment Architecture. VMM Figure 4.2

Ben Cohen
abv-sva.org



# VMM Adoption

Ben Cohen
abv-sva.org