An efficient requirements-driven and scenario-driven verification flow

Heino van Orsouw
NXP Semiconductors
Outline

• Motivation
• Verification methodologies
• Combining Requirements Based verification (RBV) and Metric-Driven Verification (MDV)
• Scenario modeling for top-level verification
• Conclusions / acknowledgement
Motivation

- Test specification setup and maintenance effort must be reduced. (x1000 spec items)
- Long and detailed specification lists weaken the overview.

Opportunities to improve:
- Efficient test specification flow must be functional verification aware.
- Coverage of IP must contribute to top-level verification quality proof.
- Automated test creation
Verification methodologies

- Directed test
  - Traditional, corner-cases, traceable
- Constraint Random
  - IP verification, UVM, automated
- Scenario-driven
  - Constraint random at top-level
  - Design intent

Top-level verification sweet spot is proper combination
RBV and MDV for top-level verification

- Requirement specs
- Test specs
- CR tests
- MDV monitors
- Directed tests
- Coverage results
- Scenario models
- Generated Test cases
- reused UVC

Specification environment
Verification environment

© Accellera Systems Initiative
Traceability relations: Requirements to Test results

**Specification environment**
e.g. IBM DOORS

- **Requirements**
  - UART
  - RS1
  - RS2
  - APB

- **Test Spec**
  - UART
  - TS1
  - TS2
  - TS3
  - APB

**Veriﬁcation environment**
e.g. Cadence vManager

- **Executable Test Spec**
  - **Descriptions**
    - UART
      - TS1
        - Directed test x
        - Checkers/assertions
        - Functional coverage Y
      - TS2
      - TS3
      - APB
  - **Goals**
    - Cumulative 100%
    - Directed test x: Pass
    - Checkers/assertions: Pass
    - Functional coverage Y: 70%

- **Metric Implementations**
  - Checkers
  - Assertions
  - Coverage

Traceability relations: Tests Refines Implements
Test Cases on TopLevel

Move from directed/use case driven to scenario driven as:

• Scenarios specify the behavior of the system
• Scenarios define the intent of the design
• Functional End Points known, but paths can vary
• End Points can be linked to Test Spec Items for early traceability!
Simple top-level scenario for UART

- Main
- SelIpClk
- SelSysClk
- UartMessageSize
- UartMode
- UartBaudRate
- Uart_Cmd
- Uart_Dbl_Cmd
- Uart_Short
- Uart_Bool
- Uart_Tx
- Uart_Rx
- Check

- OscIpClk
- PllIpClk
- OscSysClk
- PllSysClk

- Constraint
- Selected path
- Test Specification
- AND
- OR
- Constraint Random
Results of one scenario and it’s IP coverage results
Conclusions

• Executable Test Specification, is proper place to refine test specifications.
• Scenario modeling increases confidence of completion in top-level verification.
• Reuse from IP to Top-level verification can be achieved by passive Verification Component.
• This methodology concept has been applied successfully.
• The overview of verification quality and coverage has increased.
Acknowledgment

• Thanks to NXP colleagues
  – Walter Tibboel
  – Shuang Han for UVM support
  – Erwin de Kock and Martin Barnasconi for contribution in the discussions
  – Monica Farkash for helping with scenario modeling and generation

• Thanks to Cadence R&D
  – in special to Thomas Ziller for the vManager support
Questions?