Overview

- Motivation/Rationale
- Solution Insight
  - Module Level Verification
  - Sub-system Level Verification
  - System Level Verification
- Results
- Conclusions
- Next Steps
- Questions
Motivation/Rationale

- Extensive Pre-silicon Verification and Improved Design Flow
- Platform based verification environment
- IC Verification
  - How to enable support to scalable platforms for SoC verification
    • Ease of extensibility, configurable topology, processor independent env.
    • Configurable/Adaptive methodology for SoC verification
  - How to enable re-use infrastructure from Module Level to System Level Verification
  - Common Industry Standard Verification Methodology from Module Level verification to Subsystem/System level verification
Motivation/Rationale

- HW/FW/MW co-verification
  - How enable re-usability of Test Cases/SW from Pre-silicon to Post Silicon Validation?
  - How to share simulation snapshot to SW team?
  - How to enable test-case writing at System level test-case without knowing the HVL Language e.g. Software Engineer?
  - How to control execution and constraining software and Hardware Stimuli?
  - How to get functional coverage details of HW/SW API's/Tests?
  - How to Ease Debugging HW/SW for failures?
Solution Insight
Module Level Verification

- Virtual Sequencer
- vIP’s for various interfaces implemented by the DUT
- Scoreboard for data integrity checks and checkers for protocol compliance
- eManager for automation/vPlan for defining coverage goals
- Re-use of vIP’s (OVC’s) across projects/DUT’s
- Supports interoperability between different languages
- Open Verification Methodology → Industry standard methodology to achieve metrics-driven verification (MDV)
Solution Insight
Sub-System Level Verification
Solution Insight
Sub-System Level Verification

- Verification environment uses:
  - Metrics driven verification methodology using OVM.
  - Incisive Software Extensions package to control/constrain execution of Software
  - vIP’s to control/constrain HW stimuli
  - Re-use of module level vIP’s (e.g. AHB, APB, SWD etc) for protocol compliance checks
  - Scoreboard to check data integrity
  - Coverage details to get insight into the progress of the verification goals
  - Virtual sequences to co-ordinate/control test scenarios to be executed on CPU and vIP’s on different interfaces
  - eManager to support automatic regression runs
  - vPlan to capture the code coverage and functional coverage goals
Solution Insight
Sub-System Level Verification

- Functional Checks:
  - Interconnection
  - Sub-module functionality
  - Interrupt handling
  - Debug interface
  - Features of CPU
  - Clock gating features implemented in the subsystem
  - Dynamic traffic on multiple interfaces of the DUT

- Automation of regression runs using Enterprise Manager

- Coverage goals mapped to vPlan for final sign-off
Solution Insight
System Level/Pre-Silicon Verification

- System Level /Pre-Silicon Verification Environment
Solution Insight
System Level/Pre-Silicon Verification

- Scalable Verification framework
  - Extensible, Processor Independent, configurable Topology

- Test Stimulus generated using TCL (No recompilation of ENV)

- TCL scripts provide hooks to the Sequencer of vIP’s

- Re-use of the verification environment from Subsystem Level

- Convergence of Verification setup and Validation setup
  - Stimulus generated for Pre-Silicon verification re-produceable for Post-Silicon Validation
Results

- Scalable System Level Verification Environment
  - Ease of extensibility
  - Processor Independent
  - Configurable Environment (enable/disabling of verification components)

- Environment enables randomization of SW and HW stimuli
  - Methodology enables HAL API’s verification when SW application is not available

- Bottom-up verification approach
  - Module Level Verification (for new IP’s Developed)
  - Reuse the vIP’s and key verification components at System Level

- Reduced Bug propagation to System Level => Reduced Debug effort
- Re-use of infrastructure IP’s across verification scope
Conclusions

- Platform Centric Verification Approach
  - Addresses the key project requirements

- Environment is easily Scalable

- Innovative approach to enable generation of constrain randomization of SW and HW stimuli
  - Enables testing of SW API (When application SW not available)

- OVM based Metrics Driven Verification Methodology is suitable @
  - Module Level Verification
  - Subsystem Level Verification
  - System Level Verification

- No bugs propagated from Module Level (verified with MDV) to System Level

- vIP’s reusable across organisation

- Single Framework @ System Level Verification (Reduced Maintainability)
Conclusions (contd. …)

- Reduced Silicon bring-up effort (for first received samples)
- Convergence of Verification and Validation setup (Reuse SW and Stimuli)
  - Reduced development effort
- vIP’s available for Re-use to the Next Projects/across projects
Next Steps

- Migration to UVM
- Use complementary methodology/solution e.g. IFV
- Re-use of module level test cases @ System Level
- Mixed Mode simulations with digital infrastructure?
Questions
Thank you