Towards a Hybrid Verification Environment for Signal Processing SoCs

Jan Hahlbeck, Steffen Löbel, Chandana G P
NXP Semiconductors Germany GmbH
Agenda

• Motivation – Why are we using a hybrid approach?
• Introduction to our Hybrid Verification Environment
  • Walking through the six phases of our verification cycle
• Apply Hybrid Verification approach on an exemplary DUT
  • Results
• Summary / Outlook
Motivation

• Dealing with radio SoC verification in automotive domain
• Demand for combination of traditional functional verification flow with formal verification and MATLAB signal processing reference models -> Hybrid Verification
• Be prepared for further methodology shift
  • Formal Property Verification
  • Automated Formal Verification
  • Motivate engineers to try out formal related apps and techniques
• Tackle verification tasks from different angles
• Verification engineers sleep better with a formal proof!
Hybrid Verification Environment - Overview

Hybrid Verification Environment

- MATLAB Simulink
- UVM Toolkit
- Formal Toolkit
- Regression Toolkit

Phase 1.1
Structural Connectivity (CONN)

Phase 1.2
MATLAB Model Generation

Phase 2.1
Functional Testbench (UVM)

Phase 2.2
Formal Testbench (FPV)

Phase 3.1
Regression & Coverage Analysis

Phase 3.2
Unreachability Analysis (UNR)

RTL Release
MATLAB Simulink Models
Test Specifications

Verification Release
Regression Report
Quality Reports
Phase 1.1: Structural Connectivity
Phase 1.2: MATLAB Model Generation

Hybrid Verification Environment
- MATLAB Simulink

MATLAB Scripts
- MATLAB Simulink Models
- Run MATLAB Simulink Coder Models
- C/C++ Models
- MATLAB DPI Wrapper Creation
- DPI Wrappers
- DPI Reference Model Creation

DPI Reference Models

Test Specifications
- Required Accuracy
Phase 2.1: Functional Verification

- Functional Test Development
- Hybrid Verification Environment
- UVM Toolkit

- Setup / Update UVM Testbenches
- Functional Test Development
- Functional Coverage Development
- SVA Property Development
- Tests
- Covergroups
- Simulation Properties
- Common Properties

- Code Generator
- Linter Rules
- Run Testbench Linter
- Run Functional Simulation
- Quality Feedback
- Exchange with Formal Verification

- RTL Release & Reference Models
- Test Specifications
Phase 2.2: Formal Property Verification

Hybrid Verification Environment

Formal Toolkit

Formal Properties

SVA Property Development

Run Formal Prove

Setup / Update Formal Testbench

Formal Testbench

Command Scripts

Common Properties

Exchange with Functional Verification

Code Generator

RTL Release

Test Specifications

SVA Property Development

Formal Properties

Formal Testbench
Phase 3: Regression Setup, Coverage and Unreachability Analysis

Hybrid Verification Environment

Regression Toolkit

Run Regression
Simulation Database
Unreachability Analysis (UNR)
UNR Refinement
Quality Report

UVM Testbenches
Run Testbench Linter

Formal Testbenches (CONN + FPV)
Run Regression
Formal Database
Regression Database Merge
Merged Regression Database
Coverage Analysis & Specification Mapping
Regression Report

Test Specifications
Applying Hybrid Verification

Radio Signal Processing Subsystem

- AXI2APB Bridge
- Wideband Filter 0
- Wideband Filter 1
- Wideband Filter 2
- Filter Crossbar
- Smallband Filter 0
- Smallband Filter 1
- Smallband Filter 2
- Data Packetizer
- Write DMA
- Structural Connectivity
- Functional Verification
- MATLAB Reference Models
- Formal Verification
- Synchronization Unit
- Interrupt Distribution
Coherent Data Check Flow

- Wideband Filter 0
- Filter Crossbar
- Smallband Filter 0
- Data Packetizer
- Write DMA
- UVM VIP

- MATLAB DPI Model
- Full Formal Proof
- MATLAB DPI Model
- UVM Scoreboard
- Partial Formal Proof
- UVM Scoreboard
- Partial Formal Proof
- UVM VIP
Sharing Properties

- Sharing properties between UVM and FPV via System Verilog interfaces
- Limited to DUT interface signals
- Rules
  - No liveness properties
  - No usage of free variables
  - Limit number of cover properties
MATLAB DPI Wrapper

• Required accuracy provided by test specifications
• Reference value check every clock cycle
• Standardized DPI wrapper interface
  • DPI_<name>_initialize()
  • DPI_<name>_terminate()
  • DPI_<name>_reset()
  • DPI_<name>_output()
  • DPI_<name>_update()
## Results – Coverage Merge

<table>
<thead>
<tr>
<th>Module</th>
<th>Properties</th>
<th>Sim Coverage in % (*1)</th>
<th>Formal Coverage in % (*2)</th>
<th>Merged Coverage in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wideband Filter</td>
<td>2</td>
<td>75</td>
<td>1.1</td>
<td>75</td>
</tr>
<tr>
<td>Filter Crossbar</td>
<td>106</td>
<td>94</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Data Packetizer</td>
<td>150</td>
<td>65</td>
<td>75</td>
<td>88</td>
</tr>
<tr>
<td>Interrupt Distribution</td>
<td>66</td>
<td>95</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Synchronization Unit</td>
<td>50</td>
<td>73</td>
<td>90</td>
<td>95</td>
</tr>
</tbody>
</table>

*1 Branch, Expression, Statement, Toggle on interfaces
*2 Normal Precision, Reset excluded
## Results – Unreachability Analysis (UNR)

<table>
<thead>
<tr>
<th>Module</th>
<th>Coverage Pre-UNR in %</th>
<th>Coverage Post-UNR in % (*1)</th>
<th>Total UNR in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUT top level</td>
<td>76</td>
<td>82</td>
<td>6</td>
</tr>
<tr>
<td>Wideband Filter</td>
<td>75</td>
<td>82</td>
<td>7</td>
</tr>
<tr>
<td>Smallband Filter</td>
<td>84</td>
<td>87</td>
<td>3</td>
</tr>
<tr>
<td>Data Packetizer</td>
<td>88</td>
<td>88</td>
<td>0</td>
</tr>
<tr>
<td>Write DMA</td>
<td>93</td>
<td>95</td>
<td>2</td>
</tr>
</tbody>
</table>

*1 Running UNR with uninitalized start state
Summary

• We created a powerful hybrid verification setup which combines a traditional UVM based approach with powerful extensions like formal property verification, structural connectivity checks, unreachability analysis paired with integrated MATLAB reference models.

• Hybrid verification approach provides free choice of methodology for verifications teams based on verifications tasks and team experience.

• Outlook
  • Extend the usage of formal property verification (FPV)
  • Evaluate more automated formal verification apps

Questions?