Cadence® Interconnect Workbench

Cadence System-to-Silicon Verification Summit
September 2013
Heterogeneous, Multi-core SoCs

- High levels of IP integration require innovative SoC infrastructure designs
ARM® CoreLink™ 400 System IP

IP you need for your compute sub-system

- Processors
  - Cortex®-A15
  - Cortex-A7
  - Mali™-T6xx

- GIC-400 Generic Interrupt Controller
- MMU-400 System Memory Management Unit
- CCI-400 Cache Coherent Interconnect
- NIC-400 Network Interconnect
- ADB-400 AMBA® Domain Bridge
- DMC-400 Dynamic Memory Controller
Focus for performance of a path requires us to consider other masters that may influence the delay.

Hardware influences:
- Thin links, NIC-400 configuration, QoS, L2 Cache Speed, DDR Controller speed.

Scenario influences:
- Local traffic conflict, ACE-Lite Traffic, Processor Activity.

Modeling all these HW artifacts in TLM is impractical. Accurate performance analysis must therefore use cycle-accurate RTL models.
Typical Wireless ARM based SoC

- Mix of interconnect capabilities
  - Coherent, non-coherent, high-speed, low speed

- Performance of the “Memory Funnel” is key to system performance

- Verification of these complex IPs is challenging
Cadence® Interconnect Workbench
Pre-integration Cycle-accurate Performance Analysis and Verification

For Interconnect IP Integration
• Performance of use case traffic loads
• Verify configuration functionality

For SoC Integration
• Validate performance in context of IPs

Benefits
➢ Shorten performance tuning and analysis iteration loop from **days** to **hours**
➢ Reduce testbench development time from **weeks** to **hours**
Cadence® Interconnect Workbench
Automated Testbench Assembly for CoreLink 400 System IP

Architectural Information

AMBA® Designer

User Configuration

CoreLink 400 System IP RTL & IP-XACT

Interconnect Workbench Assembly

Cadence AMBA VIP Library

UVM Testbench

Testsuite

vPlan

SimVision config

Scripts
Generate Interconnect Testbench

CoreLink AMBA Designer

Interconnect Workbench
- Testbench Generation
- VIP
  - Meta-data Library

Cascaded Interconnect
- NIC-400
- CCI-400

System Development Suite
- Functional Verification Platform
  - Incisive
- Verification Computing Platform
  - Palladium XP
Generate Interconnect Testbench

CoreLink AMBA Designer

Interconnect Workbench
- Testbench Generation
- VIP
- Meta-data Library

Generate

Cascaded Interconnect

CCI-400

IC-400

System Development Suite

Functional Verification Platform
- Incisive

Verification Computing Platform
- Palladium XP

© 2013 Cadence Design Systems, Inc. All rights reserved.
Generate Interconnect Testbench

CoreLink AMBA Designer

Interconnect Workbench

Testbench Generation

VIP Meta-data Library

System Development Suite

Functional Verification Platform
Incisive

Verification Computing Platform
Palladium XP

Generated Testbench

Functional Verification Platform
Incisive

Verification Computing Platform
Palladium XP

Cascaded Interconnect

NIC-400

CCI-400

Performance Metrics

Verification Metrics

Generate Interconnect Testbench
Generated Content - Test Suite

- Part of the generated content
  - Consists of libraries of ready to run sequences + tests
  - Covers the most common test scenarios for interconnects
  - Constrained-random all through
    - Multiple invocations with different seeds will yield high coverage
  - Easily configurable (thus no need to duplicate test files)
    - Remap mode to begin with
    - Traffic profile for specific agents/all

- Will serve as the platform for performance analysis tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>single_master_single_slave</td>
<td>single path</td>
</tr>
<tr>
<td>single_master_all_slaves</td>
<td>all paths from a single master</td>
</tr>
<tr>
<td>all_masters_single_slave</td>
<td>all paths to a single slave</td>
</tr>
<tr>
<td>all_masters_all_slaves</td>
<td>all paths</td>
</tr>
<tr>
<td>all_remap_modes_single_master</td>
<td>all remap modes, from a single master</td>
</tr>
<tr>
<td>all_remap_modes_single_slave</td>
<td>all remap mode, to a single slave</td>
</tr>
<tr>
<td>all_address_boundaries</td>
<td>boundary access to all address ranges</td>
</tr>
<tr>
<td>single_slave_regs_sweep</td>
<td>all possible regs access in a single slave</td>
</tr>
<tr>
<td>single_master_regs_sweep</td>
<td>all possible regs access from a single master</td>
</tr>
<tr>
<td>all_regs_sweep</td>
<td>all possible regs accesses</td>
</tr>
</tbody>
</table>
Performance Analysis - Demonstration

- For simplicity we have chosen an AXI 8x3 matrix
  - With 4 implementations, Basic, Intermediate, Advanced and Elite

- 32-bit databases
- Read/Write issuing/acceptance capability = 1
  - Default slave

- Address Map
  - Slave 0: 0x000000..0x00FFFF
  - Slave 1: 0x220000..0x22FFFF
  - Slave 2: 0x330000..0x33FFFF
  - GPV: 0x100000..0x1FFFFF

- QoS Enabled master
- Global Programmers View (GPV) - accessible through master 0
Performance Analysis Regression - DUT vs Scenario Matrix

- Performance Analysis gets really interesting when we can analyze data from related designs and use cases.
- We have created a Performance Analysis Regression consisting of the family of 4 interconnects ‘crossed’ with the set of 4 Performance Use Cases as shown in the table below

<table>
<thead>
<tr>
<th>DUT Type</th>
<th>Scenario (Traffic Intensity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Low: Run_1, Medium: Run_2, High: Run_3, Extreme: Run_4</td>
</tr>
<tr>
<td>Intermediate</td>
<td>Low: Run_5, Medium: Run_6, High: Run_7, Extreme: Run_8</td>
</tr>
<tr>
<td>Advanced</td>
<td>Low: Run_9, Medium: Run_10, High: Run_11, Extreme: Run_12</td>
</tr>
<tr>
<td>Elite</td>
<td>Low: Run_13, Medium: Run_14, High: Run_15, Extreme: Run_16</td>
</tr>
</tbody>
</table>
Performance Analysis GUI
Performance Analysis GUI
Latency Distribution

The imported execution result

Detected Subsystems

Detected Master / Slave Agents

Current transaction in query

Transaction details pane

Graph Control Panel
Performance Analysis GUI
Latency over time
Performance Analysis GUI
Analysis results comparison
Performance Analysis GUI
Performance debugging

- List of transactions in the selected bin
- Right click to select transaction
- Selected bin

![Graph showing latency distribution split by masters]
Performance Analysis GUI
Launch Performance Debugging => Simvision

Highlight the corresponding transaction
Summary

Cadence Interconnect Workbench for SoC Interconnect Verification, Performance Analysis and Debug

- Performance measurement and Analysis for SoC Interconnect
  - Explore performance aspects across multiple simulations, multiple scenarios
    - QoS, Outstanding Transactions, Issuing Rate, etc
  - To Optimize interconnect
    - Topology, QoS Scheme, Transaction Buffer Depths, etc
  - Visualize cycle-accurate performance against a variety of scenarios
  - Assess the effect of different traffic scenarios on performance

- Automated verification of SoC Interconnect
  - Quickly configure verification environment to the interconnect
  - Run out-of-the-box tests on the generated interconnect
  - Easily update environment to verify changes

- Accelerated Debug of Complex Interconnects
  - Show related transactions across cascaded interconnects
  - Easily follow transaction flows through complex interconnects
  - Easily identify performance outliers and quickly debug them