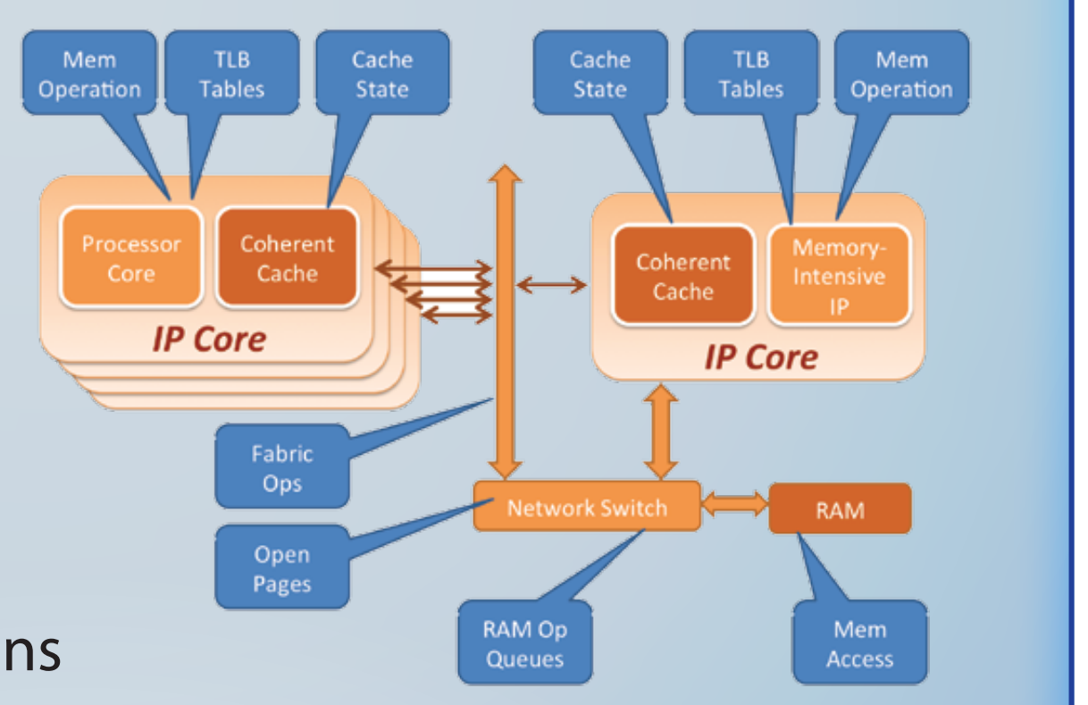
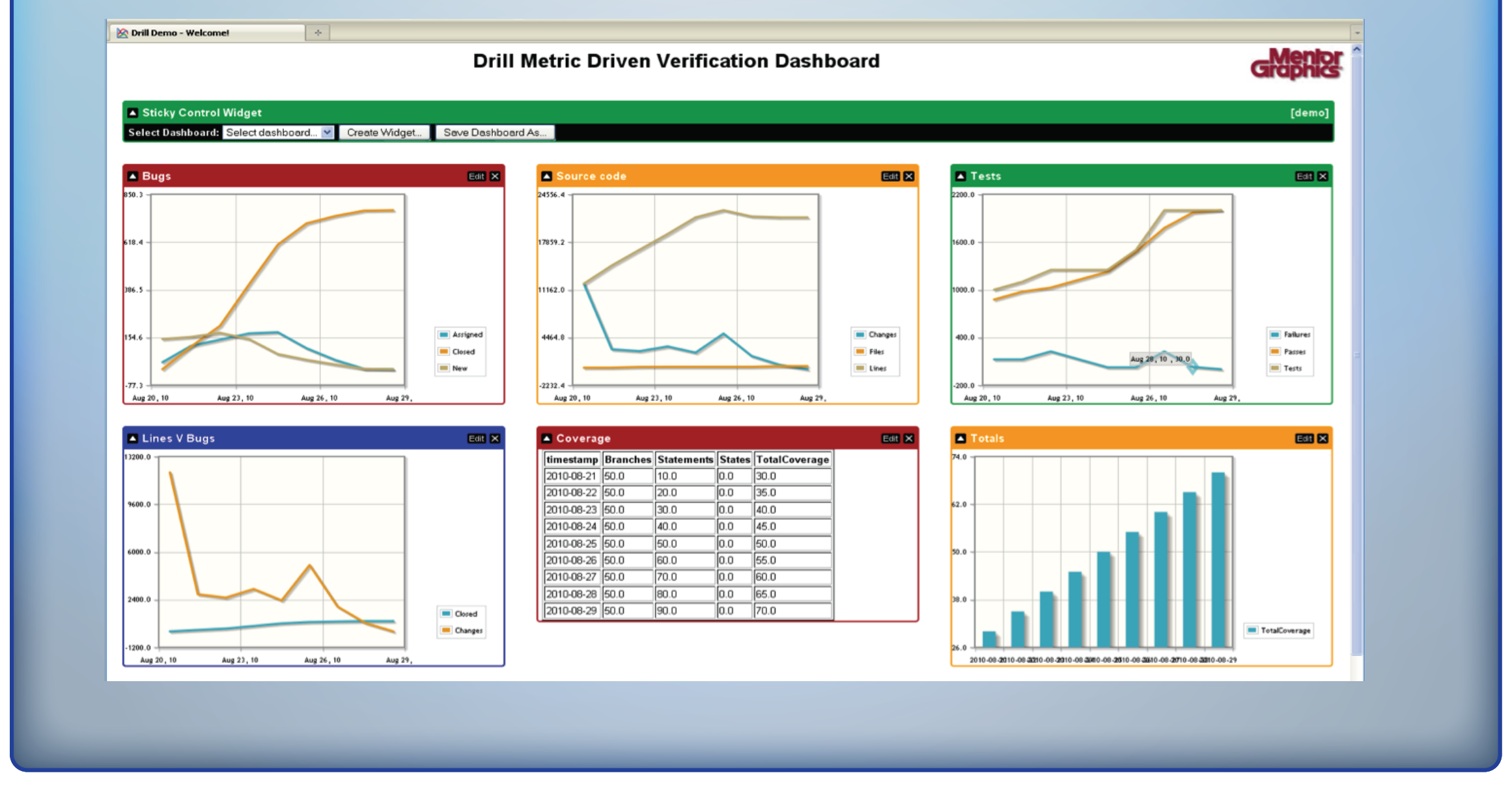
# So you think you have good Stimulus

## System-level distributed metrics analysis and results

## Driving Forces

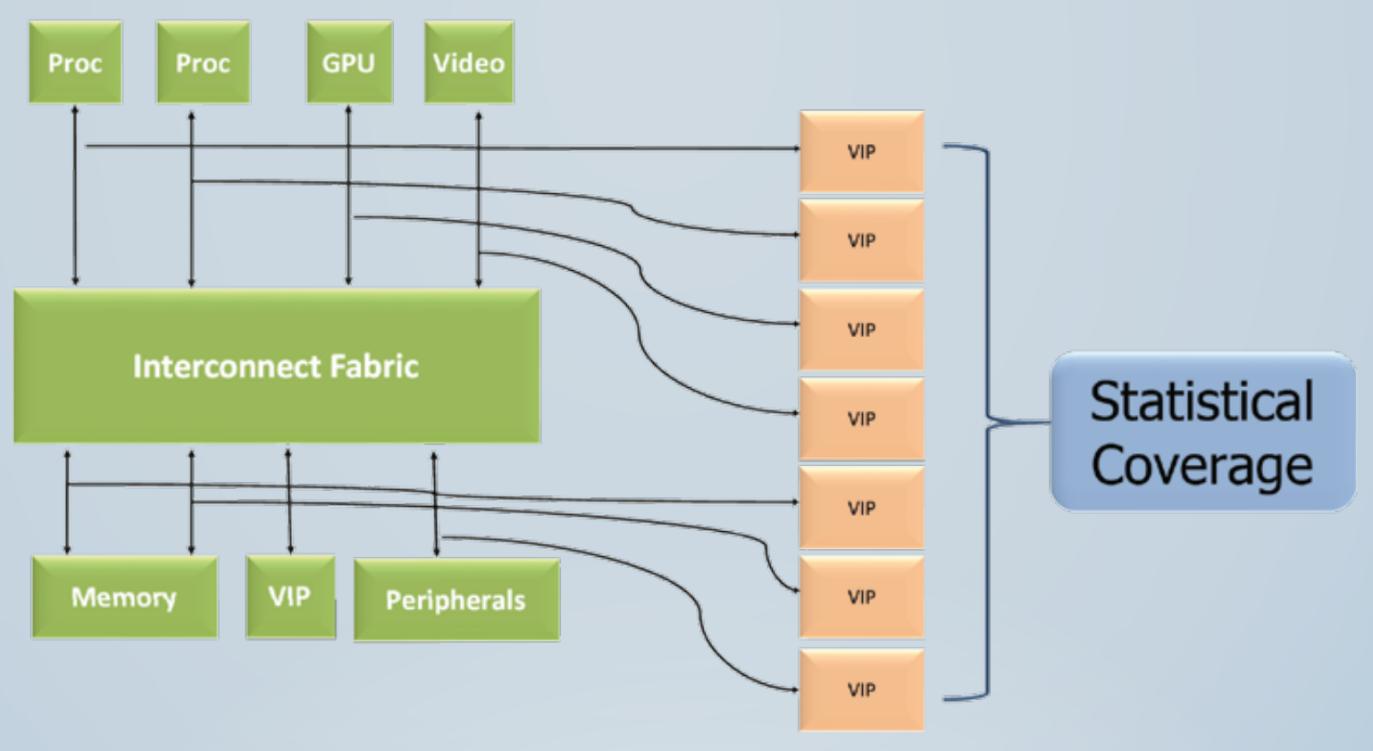


Large SoC Integrations
Complex Component Interactions
Current Coverage Metrics
Constrained Random Stimulus
Visibility across SoC required



| timestamp | Branches | Statements | States | TotalCoverage |
|---|---|---|---|---|
| 2010-08-21 | 50.0 | 10.0 | 0.0 | 30.0 |
| 2010-08-22 | 50.0 | 20.0 | 0.0 | 35.0 |
| 2010-08-23 | 50.0 | 30.0 | 0.0 | 40.0 |
| 2010-08-24 | 50.0 | 40.0 | 0.0 | 45.0 |
| 2010-08-25 | 50.0 | 50.0 | 0.0 | 50.0 |
| 2010-08-26 | 50.0 | 60.0 | 0.0 | 55.0 |
| 2010-08-27 | 50.0 | 70.0 | 0.0 | 60.0 |
| 2010-08-28 | 50.0 | 80.0 | 0.0 | 65.0 |
| 2010-08-29 | 50.0 | 90.0 | 0.0 | 70.0 |

## Method



Capture transaction-level information across system
    Cache operations
    Fabric operations
    Memory accesses
    GPU / PCIe accesses

Correlate indendent transactions
    Track operations moving between IPs
    Group transactions for analysis
    Understand system-level interaction

Analyze interactions
    Find patterns of operations
    Examine interactions between groups of transactions
    Find system-level holes in operations

Plot results
    Visualization helps understanding
    Outliers and behaviors are identified

## Project Overview

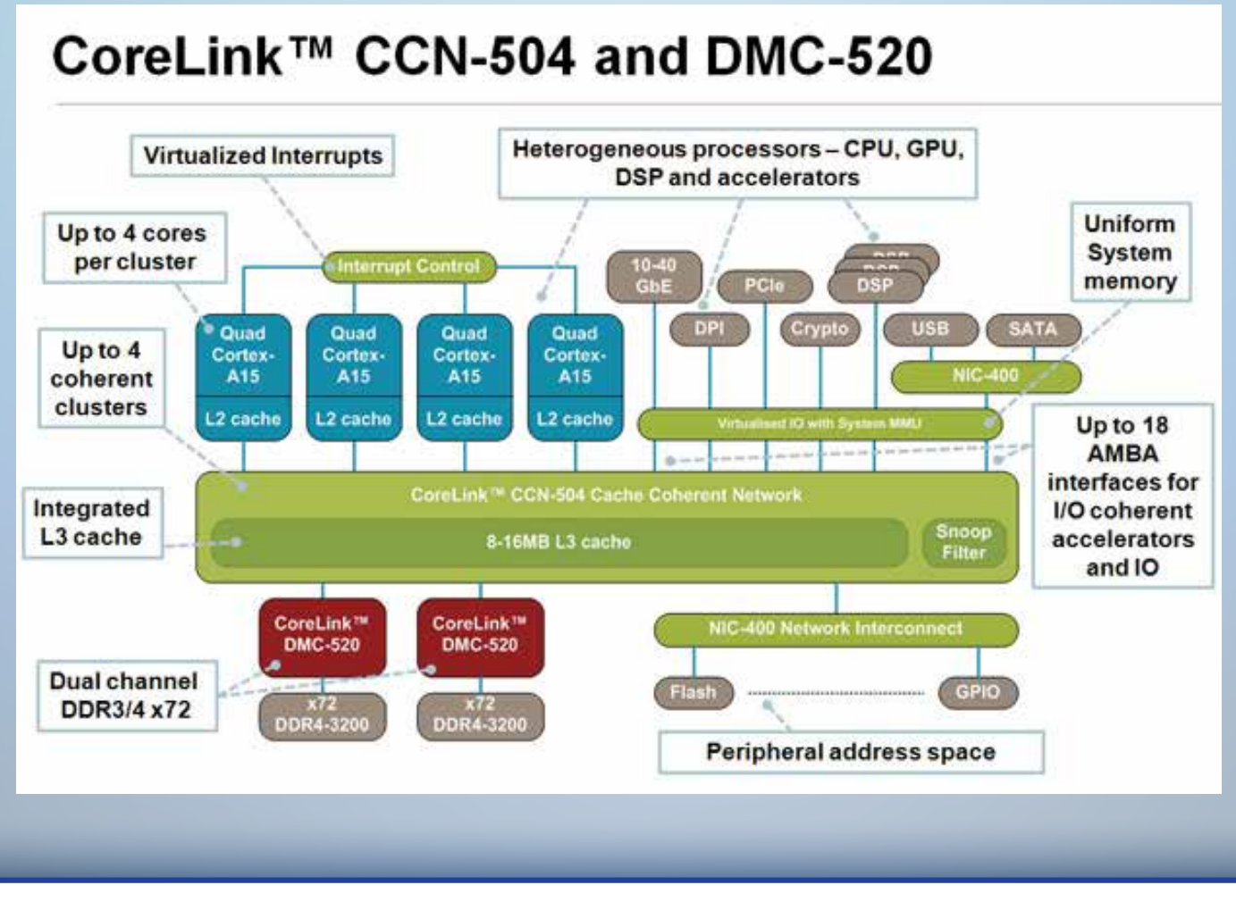
Using a current dual-cluster ARM SoC project
    Modern SystemVerilog verification flow
    High quality constrained-random stimulus
    Achieved code and functional coverage closure

Statistical coverage installed in existing project
    Measure bus, cache operations
    Traffic and interactions between IP blocks
    Running existing stimulus suite

Obtain results on entire regression suite
    Statistical - addresses, time, tests abstracted away
    Cumulative - can see if events every happened

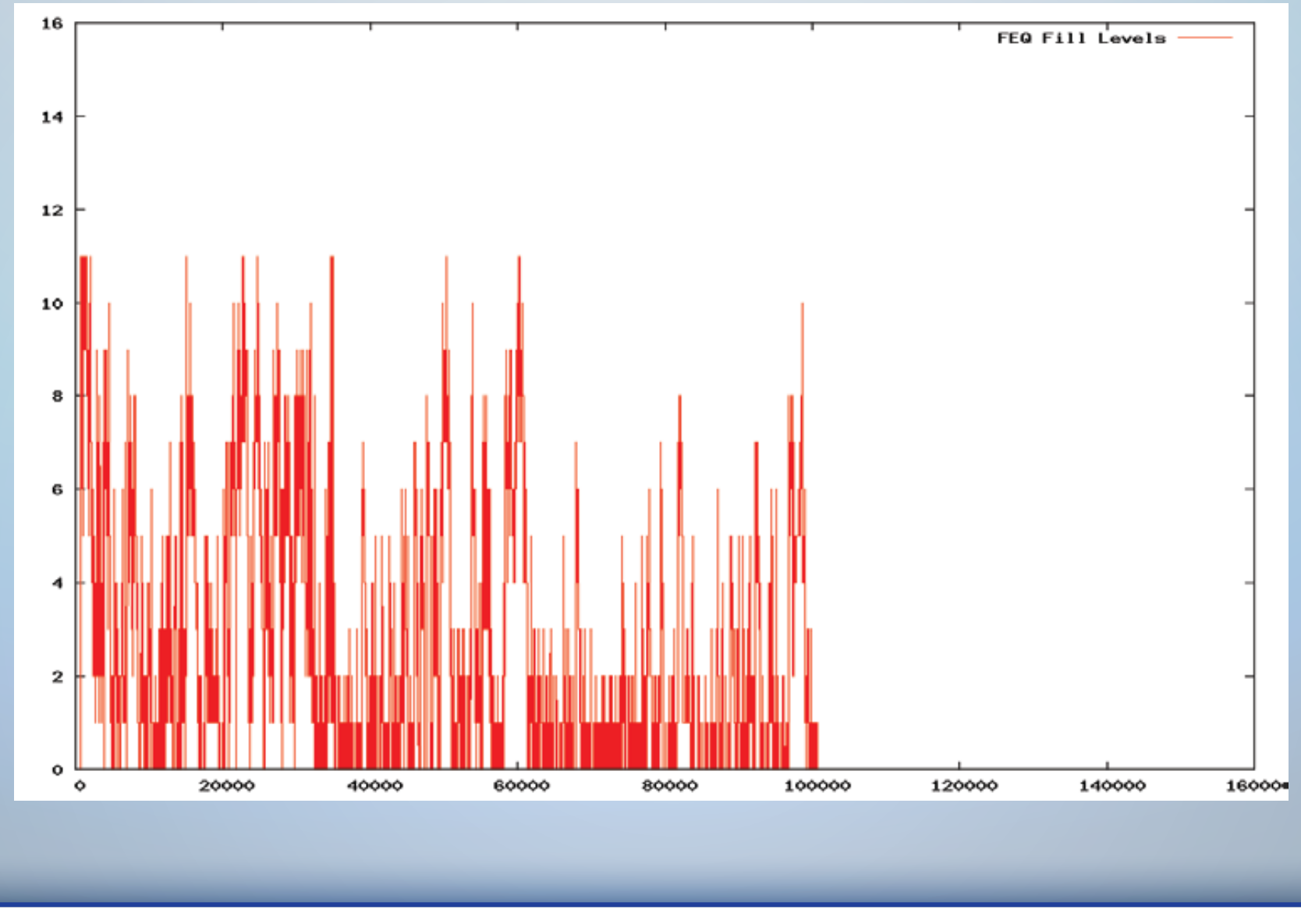
Results are based on an entire regression suite

### Representative System


CoreLink™ CCN-504 and DMC-520
Virtualized Interrupts
Heterogeneous processors – CPU, GPU, DSP and accelerators
Up to 4 cores per cluster
Uniform System memory
Interrupt Control
10-40 GbE
PCIe
DSP
DPI
Crypto
USB
SATA
NIC-400
Up to 4 coherent clusters
Quad Cortex-A15
L2 cache
Virtualised IO with System MMU
Up to 18 AMBA interfaces for I/O coherent accelerators and IO
Integrated L3 cache
CoreLink™ CCN-504 Cache Coherent Network
8-16MB L3 cache
Snoop Filter
CoreLink™ DMC-520
NIC-400 Network Interconnect
Dual channel DDR3/4 x72
x72 DDR4-3200
Flash
GPIO
Peripheral address space

## First Results

New visibility into simulation activity
- Examine activity within a simulation that is orthogonal traditional coverage and correctness checks
- Found errors in contstraints that were trivial once seen, but passed all checks
- Immediate simulation efficiency improvement

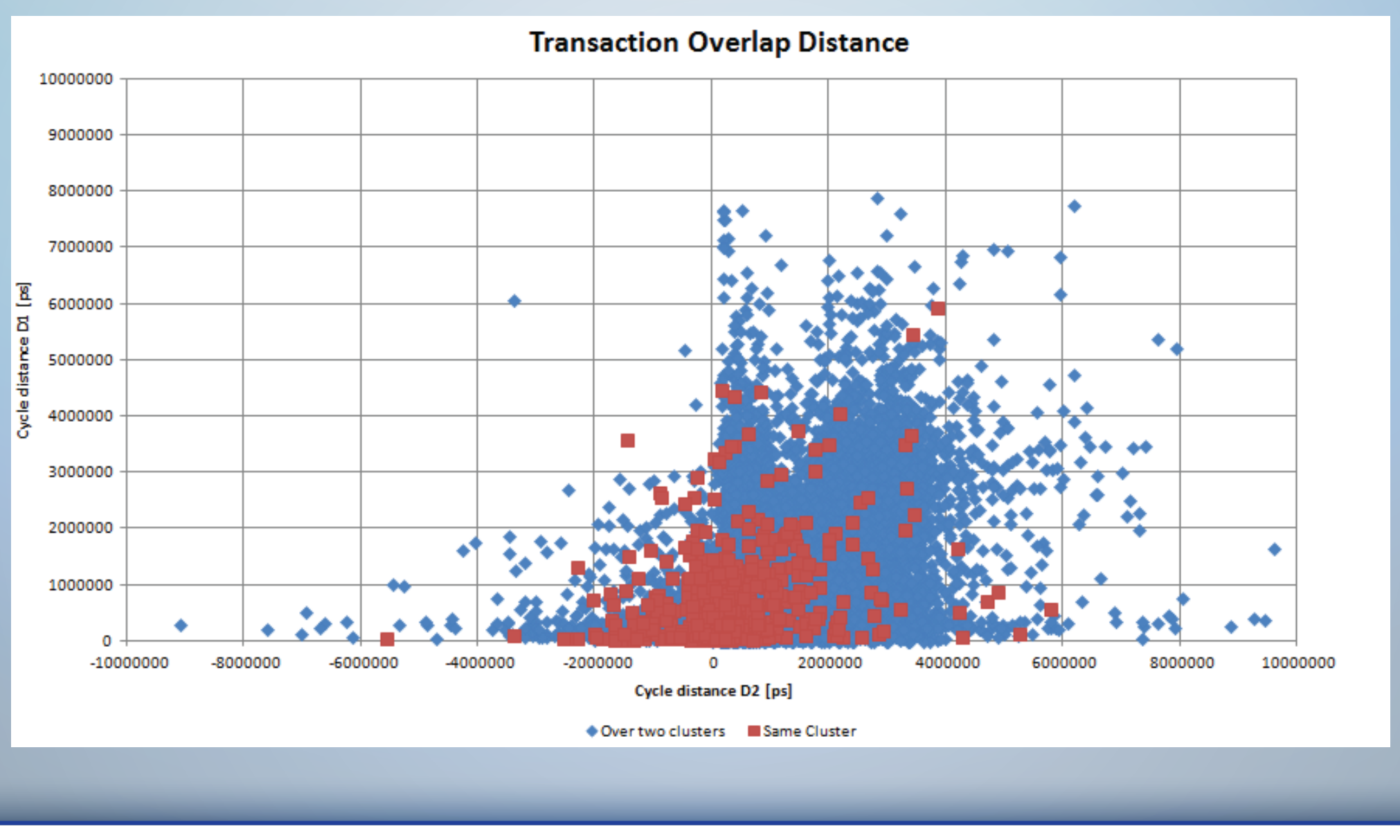


## Statistical Coverage

Finding relationships between disconnected sets of data
- Could be separated in time
- Could be from different parts of the design
- Identify and report interactions across data

Correlations may exist beyond a single simulation
- Between simulations in a regression
- Across engines or abstractions

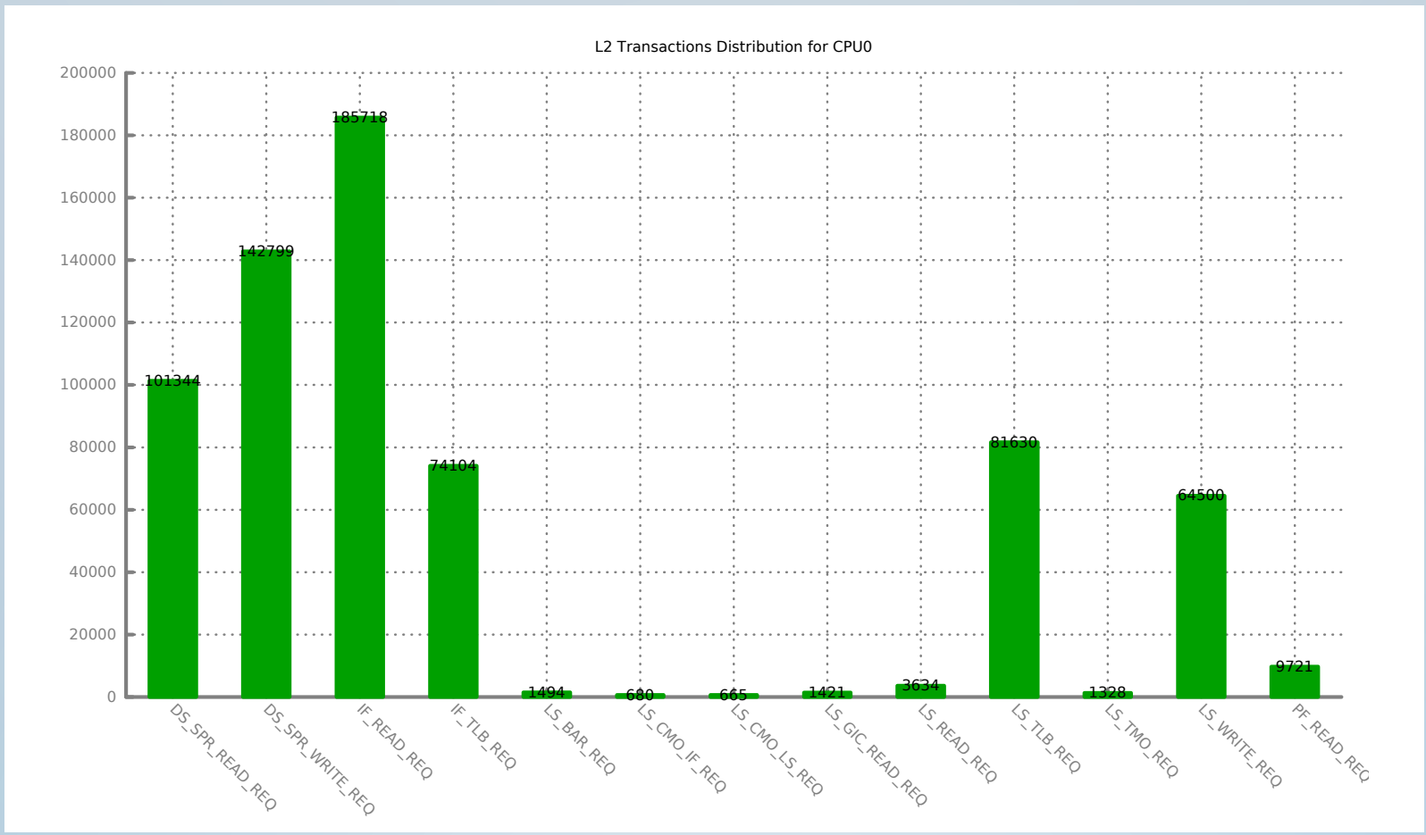
Finding and report on relationships that may be independent of test pass/fail criteria
- Fairness
- Arbitration
- Priorities



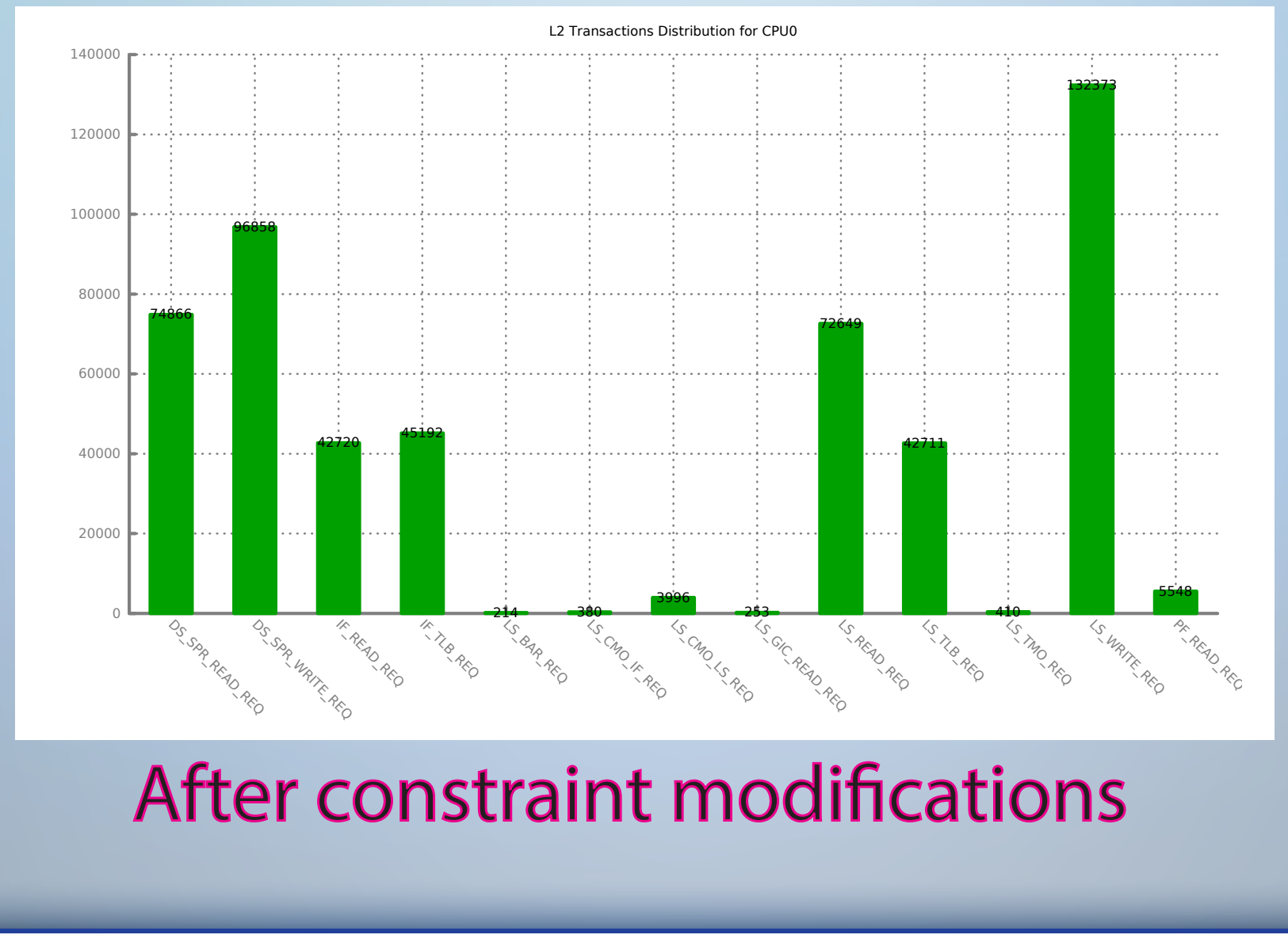
## Cache Operation Distribution

Provide a measurement of processor stimulus
- What is stimulus actually doing in the system
- How is the system reacting to stimulus
- Does the distribution match expectations

Initial stimulus distribution was not what was desired
- Functional coverage looks fine, tests all pass
- Once distribution is seen, it is possible to change constraints



Before constraint modifications



After constraint modifications

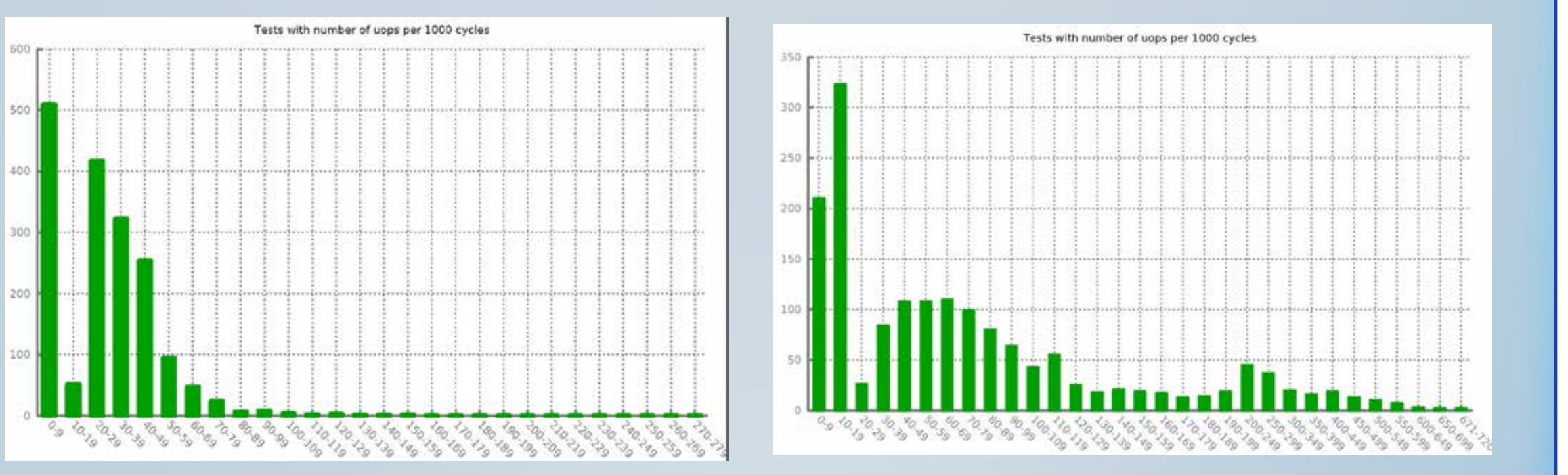
Alan Hunter, ARM, Austin TX
Andreas Meyer, Mentor Graphics, Wilsonville, OR

## Distributions

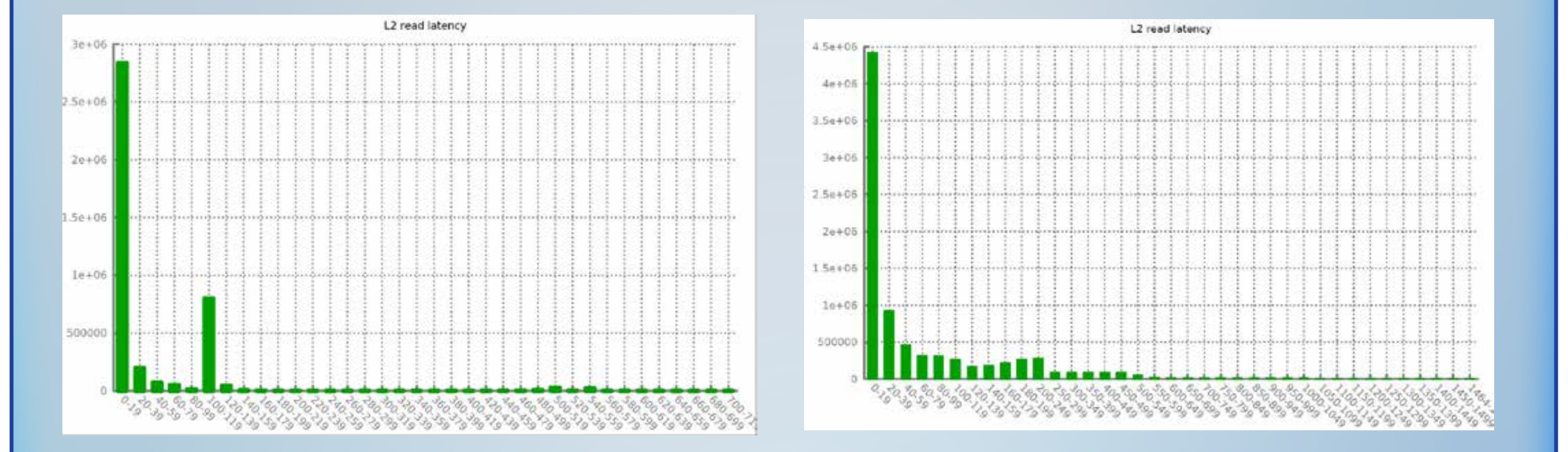
In a complex SoC, the distribution of operations in one component is significantly driven by surrounding components.

Achieving a desired distribution may require modification of constraints and configuration of several other blocks within the design.

Examples below required  a number of constraint and simulation configuration changes to achieve better distributions



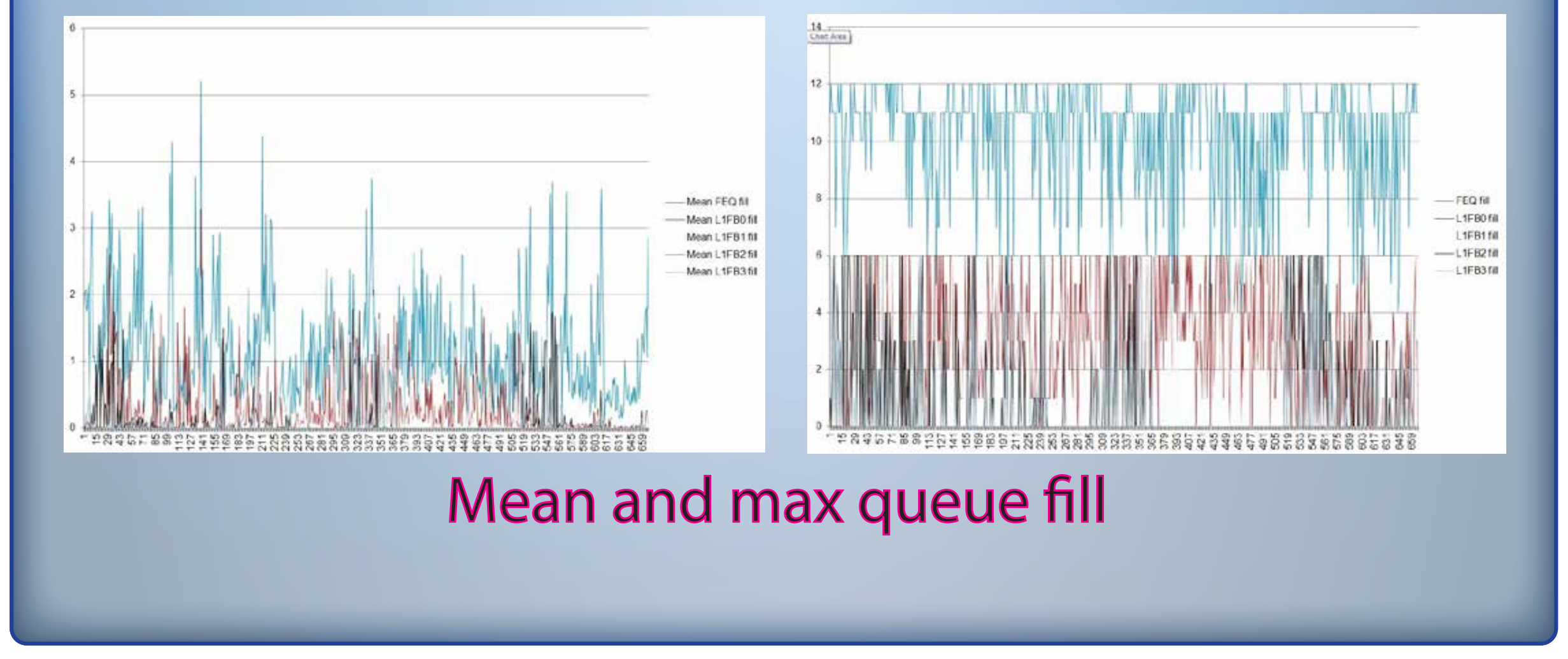
Operations per time slice before and after



Read latency distribution before and after

In some cases, even with significantly tuned stimulus, it may not be possible to reach some expected states once an IP block is integrated in an environment with other blocks.

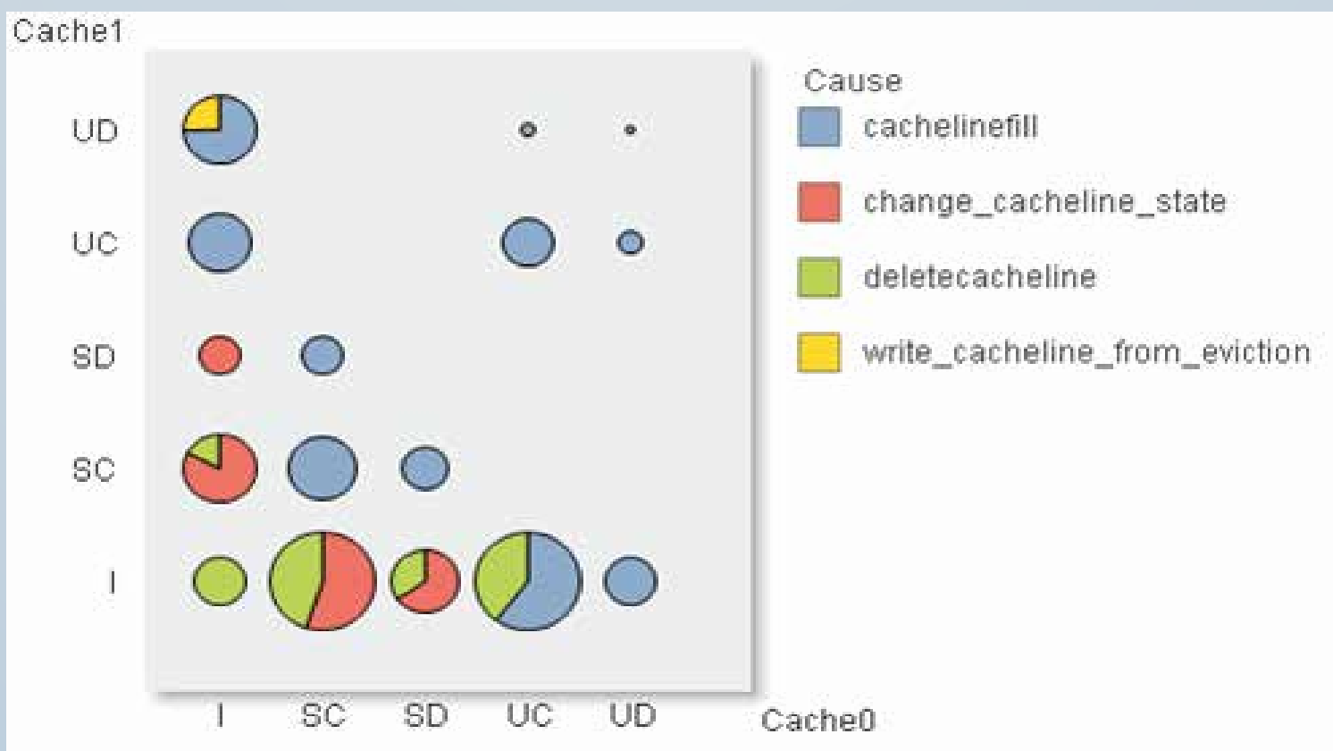
For example, a behavioral model, such as a RAM, may have been configured in a way that prevents the IP from reaching full or empty states, or a bus may have a theoretical bandwidth that a particular IP cannot achieve, meaning that corner conditions cannot be reached within that design.



Mean and max queue fill

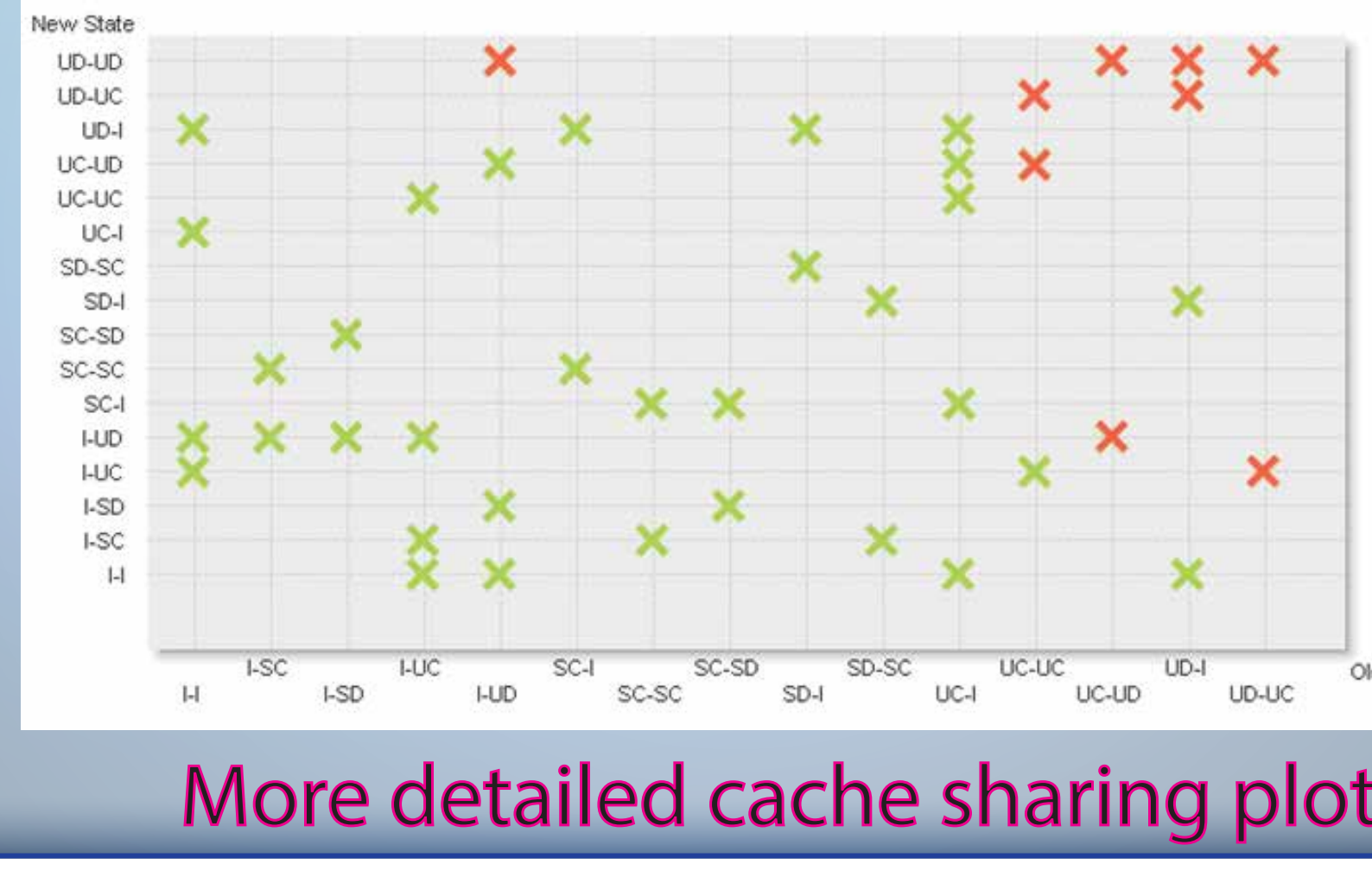
## Abstraction & Statistical Coverage

Examine patterns across large data sets
Trends in time
- Complex correlations across system
- Abstraction allows easier grasp of activity
- Many aspects of operations are dropped



Abstract cache sharing plot

Abstraction can show high-level trends
Abstraction can also hide critical issues
    - Patterns can look good at high level
    but hide low-level issues
Correct abstraction level is critical



More detailed cache sharing plot