CDN LIVE
Cadence User Conference 2012
India, October 10 - 11
connect. share. inspire.

Verifying big.LITTLE using the Palladium XP

Deepak Venkatesan
Murtaza Johar
ARM India

Agenda

PART 1 – big.LITTLE overview

- What is big.LITTLE?
- ARM Functional verification methodology
- System Validation
- System test bench – “Systembench”

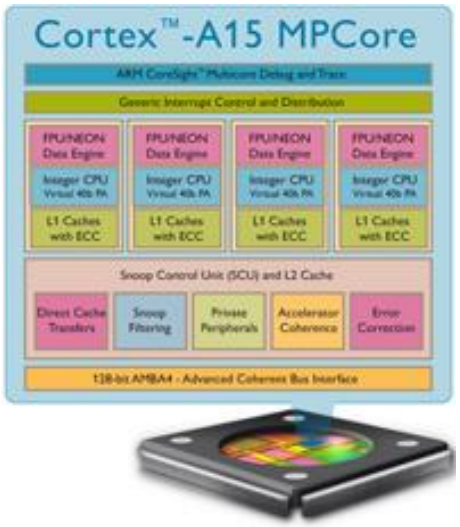
PART 2 – Palladium XP Use model

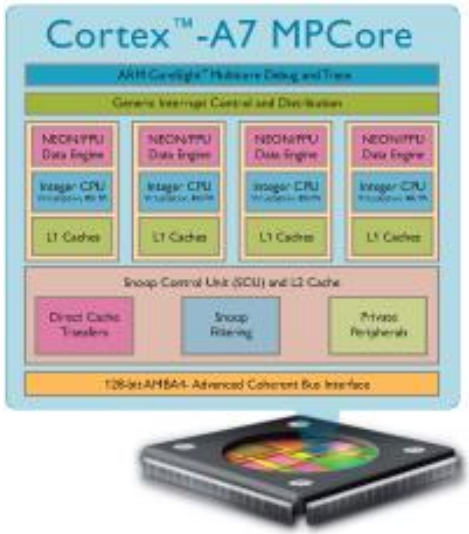
- Palladium XP features used
- Results/statistics
- Future opportunities

PART 1

big.LITTLE overview

What is big.LITTLE?





Big-Core

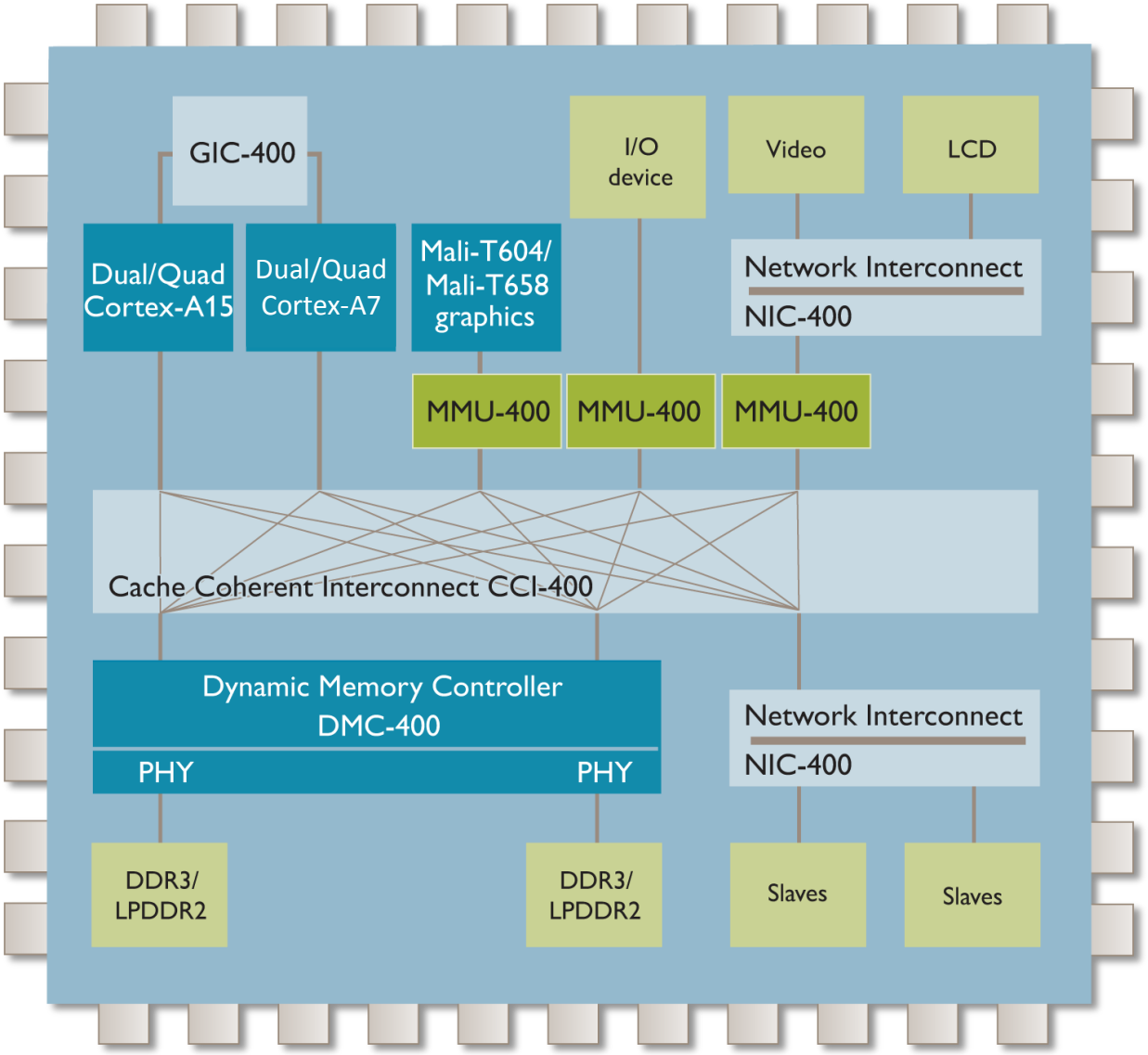
Little-Core

High-Performance for compute intensive applications

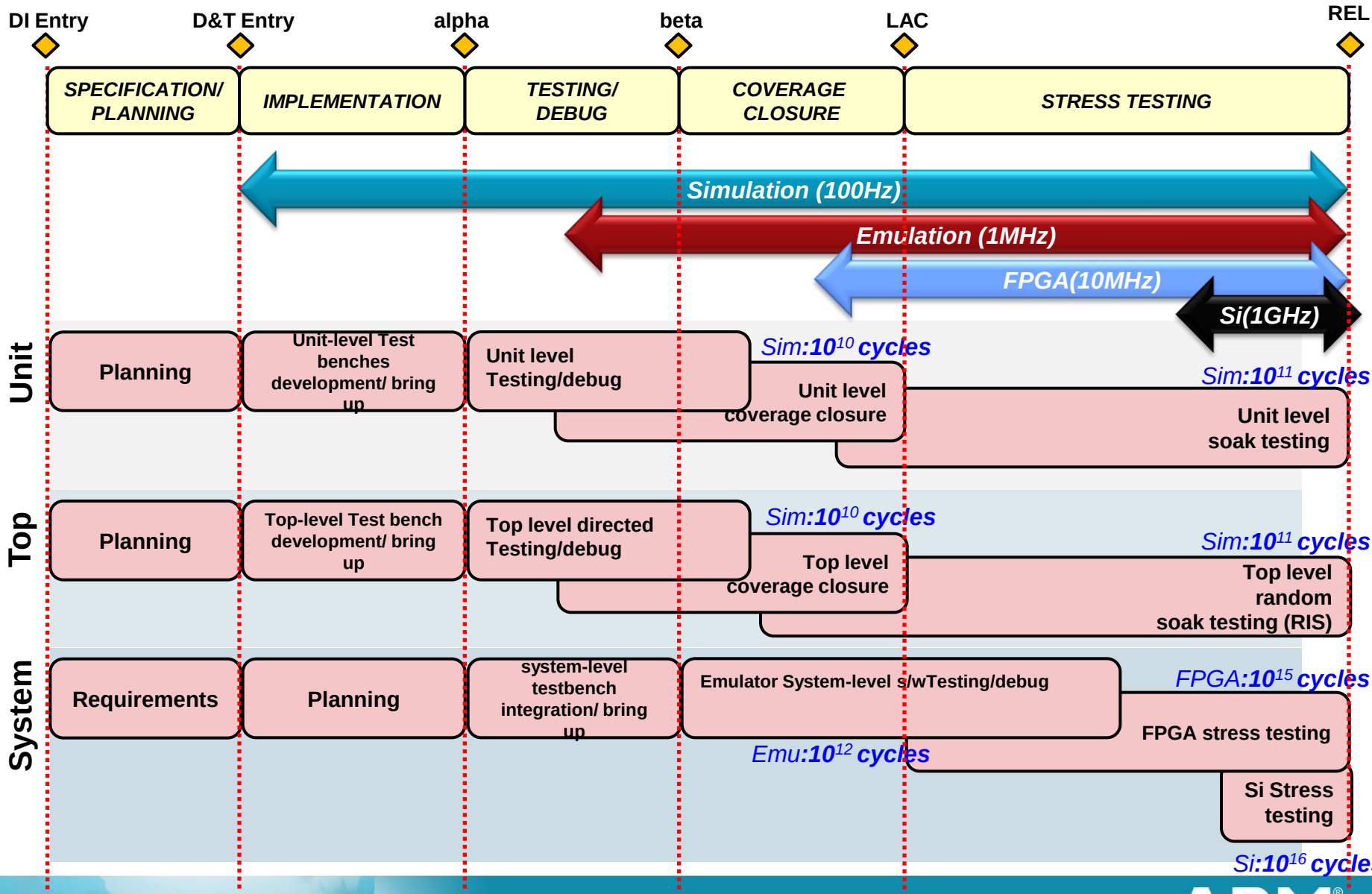
Low Power execution of majority workloads

big.LITTLE Platform Example

CoreLink™ Intelligent System IP by ARM



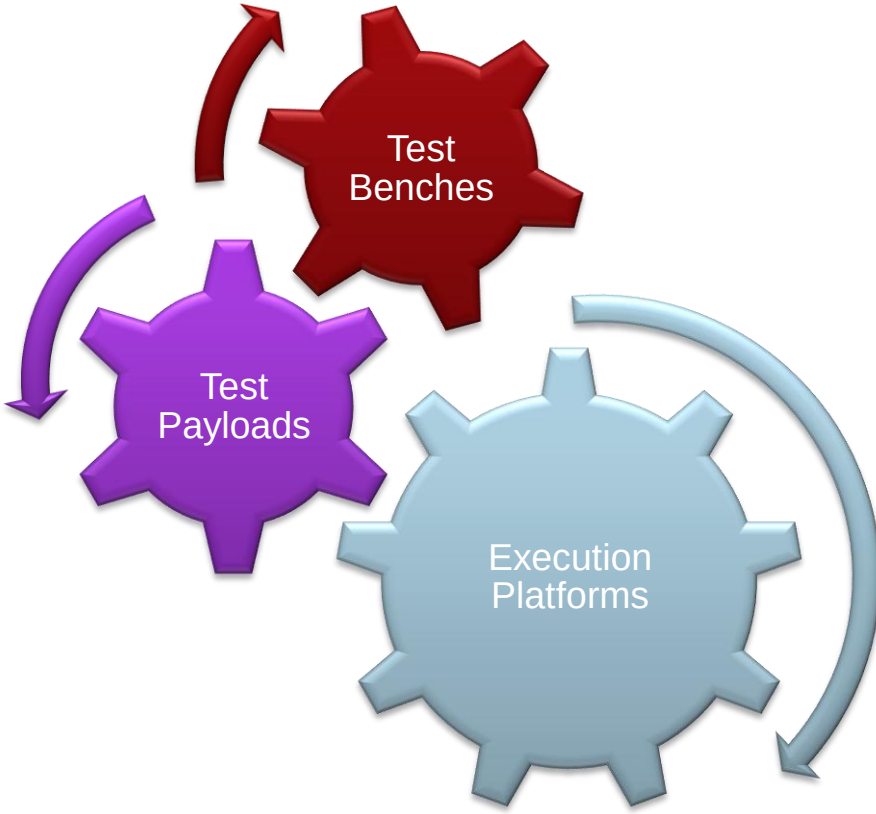
ARM Functional Verification phases



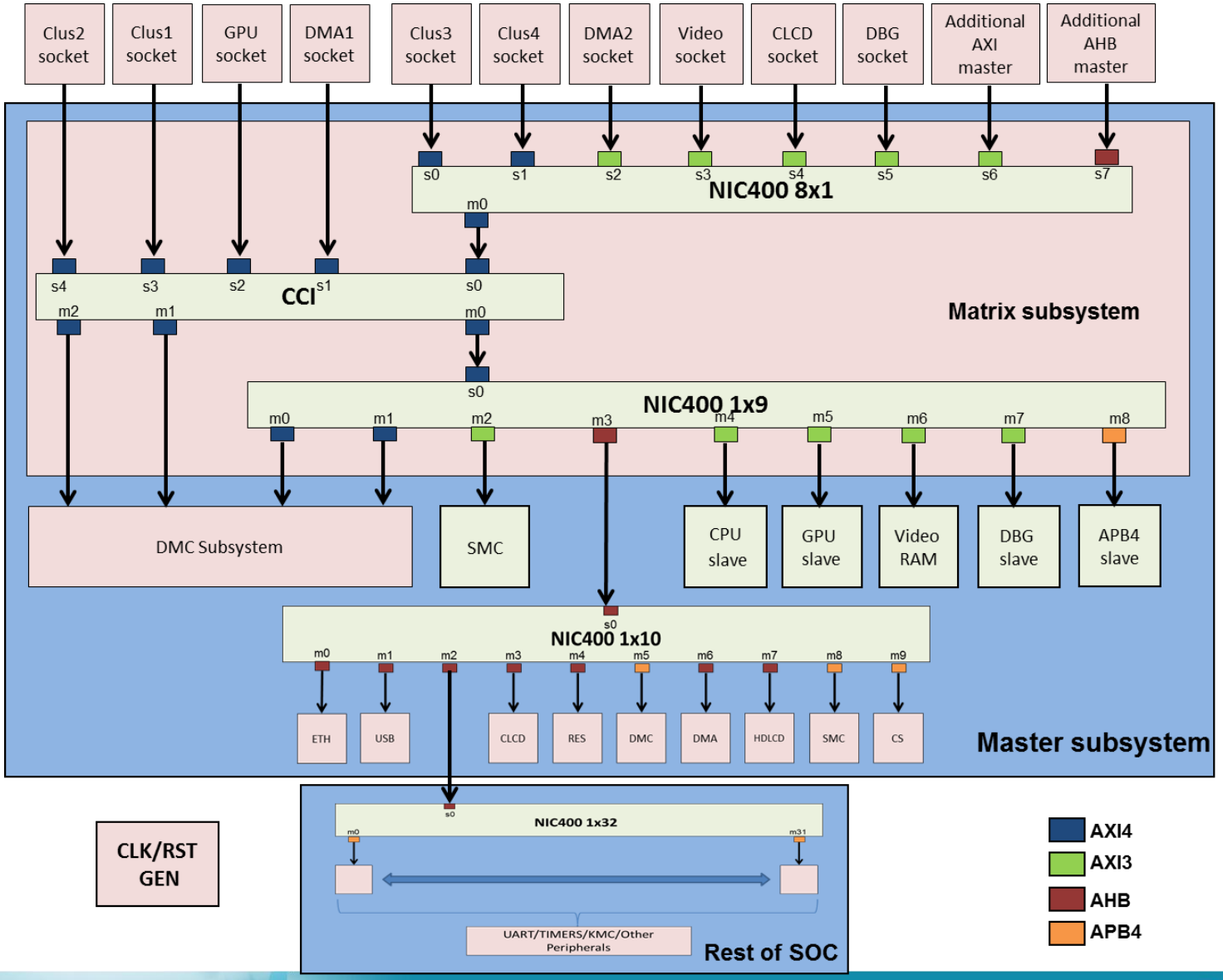
(timelines not to scale)

ARM System-level Validation

- Perform “in-system” validation of ARM IPs
  - Find IP Product bugs from real-world testing
  - (This is not the same as traditional SOC validation approach)
- To do this:
  - Build configurable System test bench
  - Support Emulation and FPGA systems.
  - Payload generation tools for stress testing
  - Plus many supporting automation flows and infrastructure



System testbench – The “Systembench”



PART 2

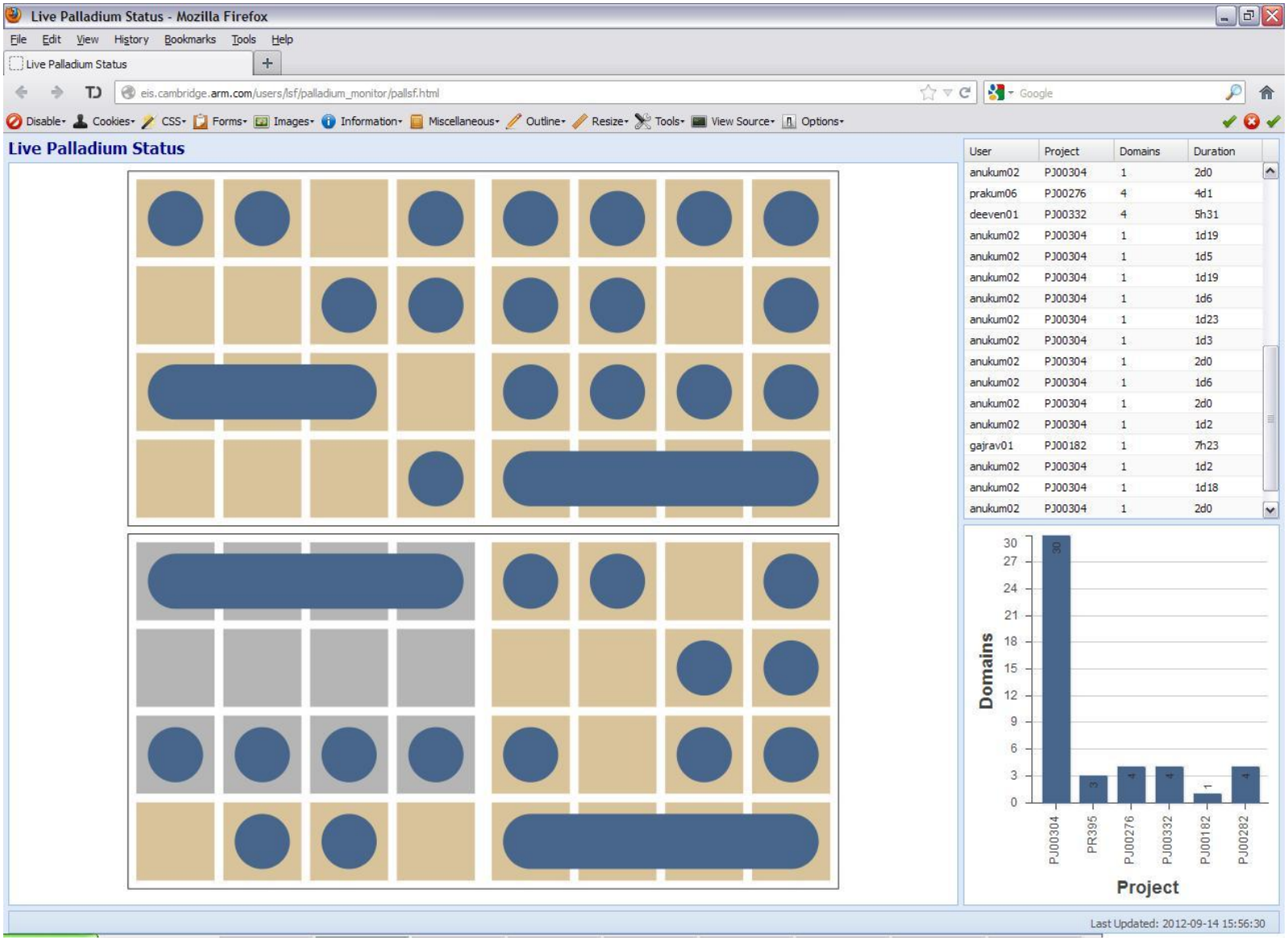
Palladium XP Use model

Palladium XP use model

- Predominantly used as stress testing platform
  - Stress mostly from multiple IP configurations and payloads
- Also used to debug failures from other platforms (e.g. FPGA)
  - Full vision mode – complete design visibility
- Other PXP features used for software analysis and qualification
- LSF Scheduler built over the PXP for scheduling multiple different jobs
  - Utilize domains effectively
  - Allow multiple users/designs/capacities to run simultaneously

- Main objective – Effectively use an expensive resource ☺

PXP LSF Scheduler widget



SDL Triggers

- Run-time debug feature
- CPU instruction/register/memory trace between two defined clock cycles
- Dumping CPU/other IP waveforms between two defined clock cycles
- Mechanism for detecting CPU hardware deadlock using probes from the test bench
  - Exit runs when hangs
- Triggering the end of test by monitoring a hardware register
- Getting dump of memory or CPU caches at known time intervals

Assertions

- Run with all assertions in CPUs, interconnect and other ARM IPs
  - Mostly OVLs and some SVAs
- AXI bus protocol checkers and violation detectors
- Test bench assertions to debug error scenarios in CPU and other ARM IP

Others

- Save and Restore
  - Save states at periodic intervals for long runs
  - Restore multiple times with additional debug hooks
- System Verilog Functional Coverage
  - Payload qualification for determining stress levels
  - E.g. number of snoop transactions
- ISS Compare in IXCOM
  - Compare Instruction executions between model and RTL

Results

- Bugs found >20
  - ~8 CAT A bugs
- PXP cycles – ~1 trillion per week during maturity phase
  - 30% on big.LITTLE
- Number of CCI transactions > 14 Billion
  - 60% on big.LITTLE
- Compile times - ~30 mins
- PXP Statistics:

CPU Cluster-1	CPU Cluster-2	System Gate count	Palladium XP frequency	Palladium XP domains
Cortex A7 MP2	Cortex A7 MP2	13 million	1.33 MHz	4 domains
Cortex A15 MP4	Cortex A7 MP4	28.5 million	1.13 MHz	8 domains
Cortex A15 MP4	Cortex A15 MP4	41.4 million	1.02 MHz	11 domains

Challenges

- System Verilog support
- Fitting multiple big.LITTLE runs
  - Very few configurations tested with GPUs
- Support for Verilog2001 features
  - e.g. generate statements
- Initial hardware reliability issues

Future opportunities

- More SV functional coverage integration
  - Port unit level to system level
  - Add system-level instrumentation
  - Use UXE 12.1 for some crucial improvements
- IXCOM migration
  - Improve speed issues
- Asynchronous clocking
- Power aware verification
- 64-bit CPU System validation

Questions?

Thank You