Experience from the Field – Challenges of Building The AXM55XX Architecture Performance Model

Storage. Networking. Accelerated.™

Ali Poursepanj, Ph.D.
System Architecture
LSI Networking Components Group
Austin, Texas

Design Automation Conference
Austin, Texas
June 3, 2013

Ali Poursepanj

Email: ali.poursepanj@lsi.com **Phone:** 512-821-6734

- Ali is a Distinguished Engineer and a lead system modeling engineer in the System Architecture Group at the LSI Network Computing Division in Austin, Texas.
- He has more than 20 years of experience in processors, systems and telecom SoC performance modeling and analysis. Prior to LSI, he worked as an architect and lead modeling engineer at Freescale, AMD, IBM, and Motorola. He also taught college part-time.
- Ali obtained his Ph.D. in Electrical and Computer Engineering from the University of Texas at Austin.
- He has published numerous papers in the area of performance modeling and workload characterization in IEEE, ACM, and EE Times publications.
- Ali has been a representative of several companies in standards committees, such as Power.org, Network Processor Forum (NPF), and EEMBC.
- He is interested in the system architecture performance modeling of layer 1, 2, and 3; computer architecture; workload characterization and modeling; highly accurate models; fast and accurate simulation models; reusable IPs; and software engineering.

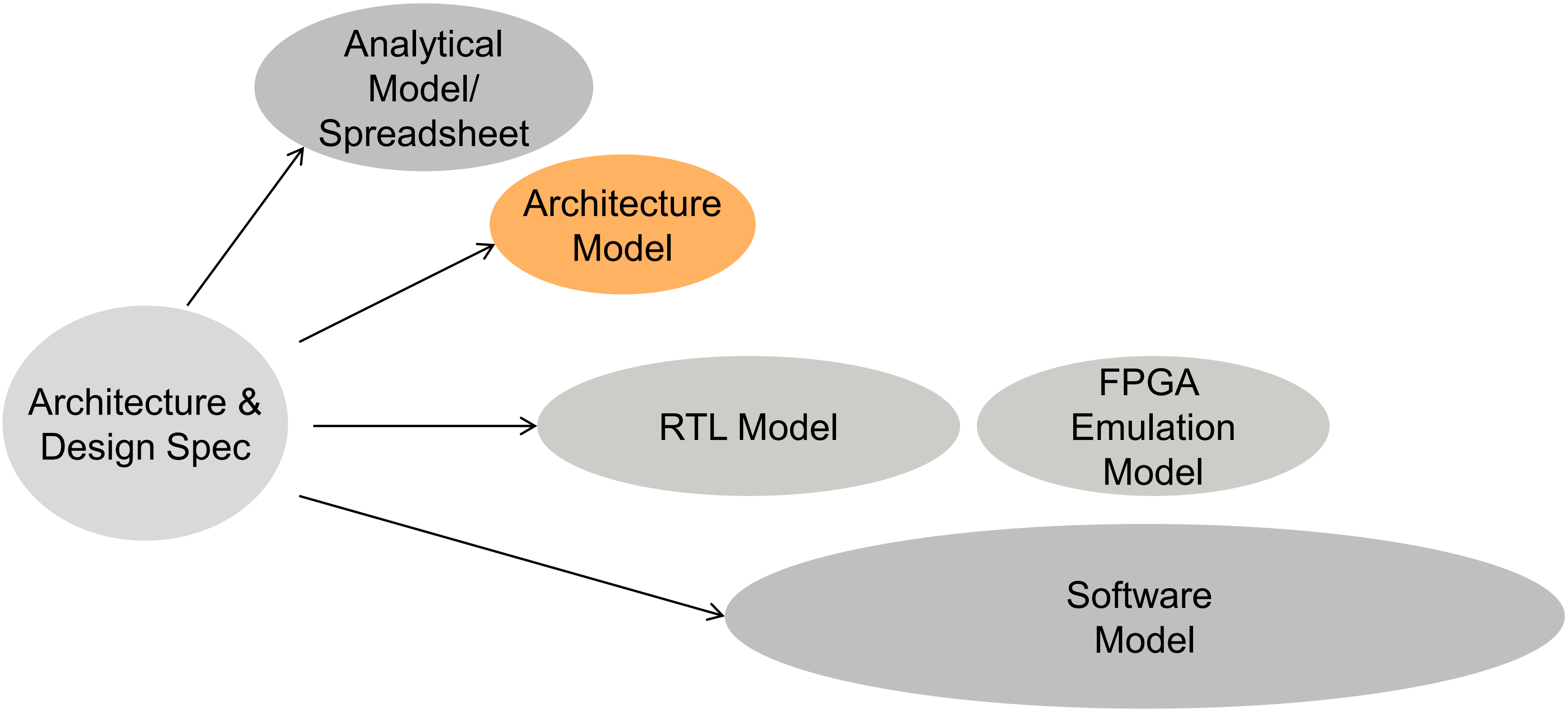
Abstract

The LSI Axxia 55XX SoC is a family of communication processors for multi-radio base stations wireless networks, combining LSI's specialized networking engines and Virtual Pipeline Technology with ARM's cores and cache-coherent interconnect. This paper describes the challenges faced in the performance modeling and analysis of such a complex SoC, including: a) the development of cache-coherent, I/O coherent, and non-coherent, cache-hierarchy aware use cases and traffic models, b) the development of a simple, address-aware, cache-coherent CPU cluster, c) the integration of IPs/performance models from multiple vendors in a common TLM2.0 based SystemC environment

Agenda

- Overview
- Challenges of architecture model development
  - Workload models
  - Hardware models
  - Third-party IP integration
- AXM55XX Performance Model
- Summary

Model Evolution



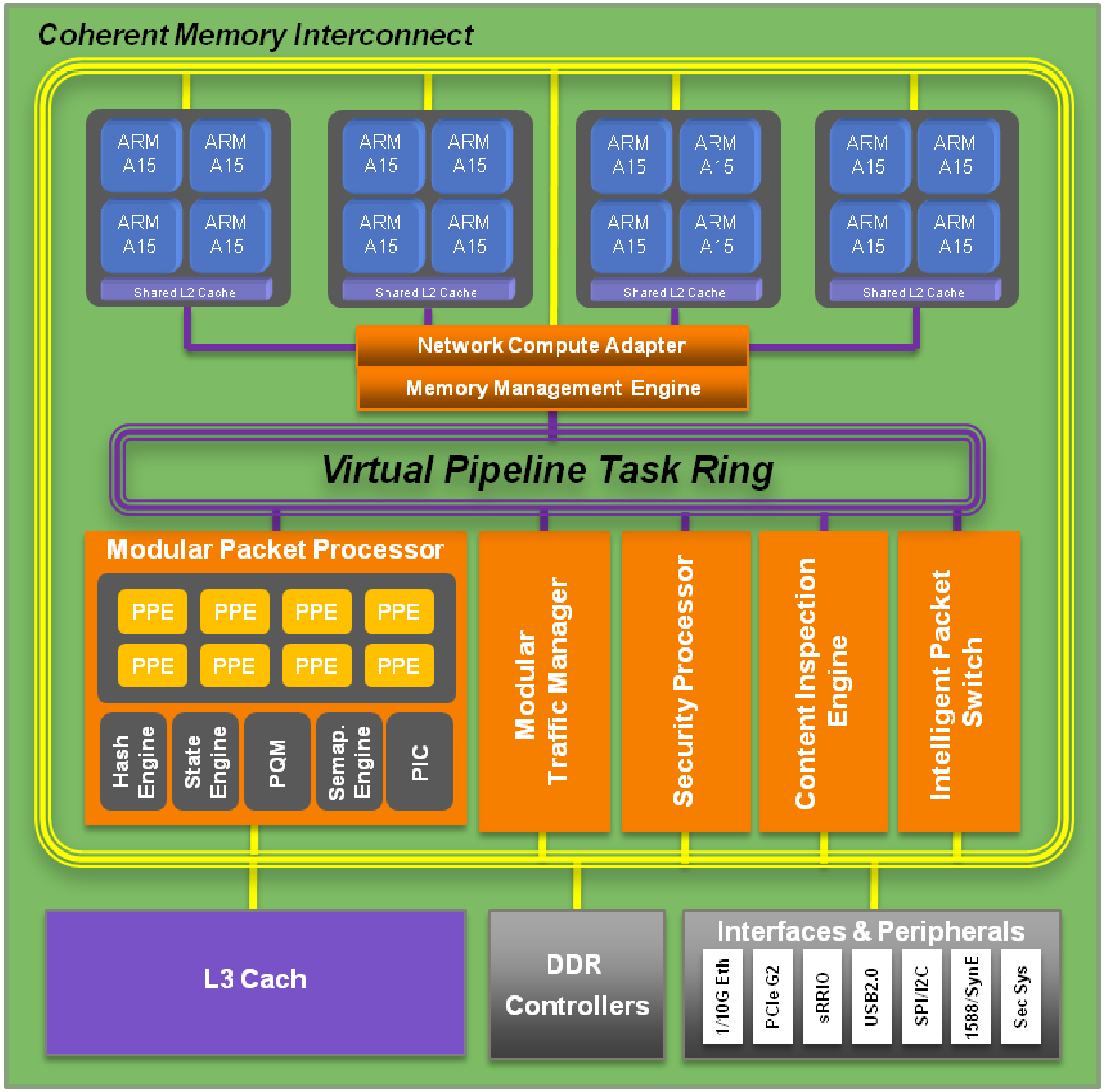
Architecture Models

- Design exploration
- Performance projection
- Design trade-off analysis
- Architecture validation
- System performance analysis

Challenges of Building Architecture Model Development

- Accurate application models
- Accurate hardware models
- Dealing with third-party IPs
    - Technical
    - Legal
    - Cost

AXM5516 Block Diagram



AXM5500 Performance Model

- OSCI SystemC simulation environment
- Used legacy and third-party IPs
- Used TLM 2.0 sockets to connect IPs
- Some of the IPs were carbonized
- Workload models were generated for several use cases
- Legal paperwork created some delays for delivering the model to some customers
- The model supported coherent interconnect and snoop transactions

AXM550 Model Traffic Generators

- Smart traffic generators were used to model the memory accesses from engines and cores.
- Request payloads included: id, address, type, priority, request size, time stamps.
- We used both generic and specialized protocols for connecting the IPs together.
- We used both in-house and third party IPs
- The traffic generators were cache and snoop aware. That required modeling local caches and snoop interaction among coherent interconnect, CPUs and devices.

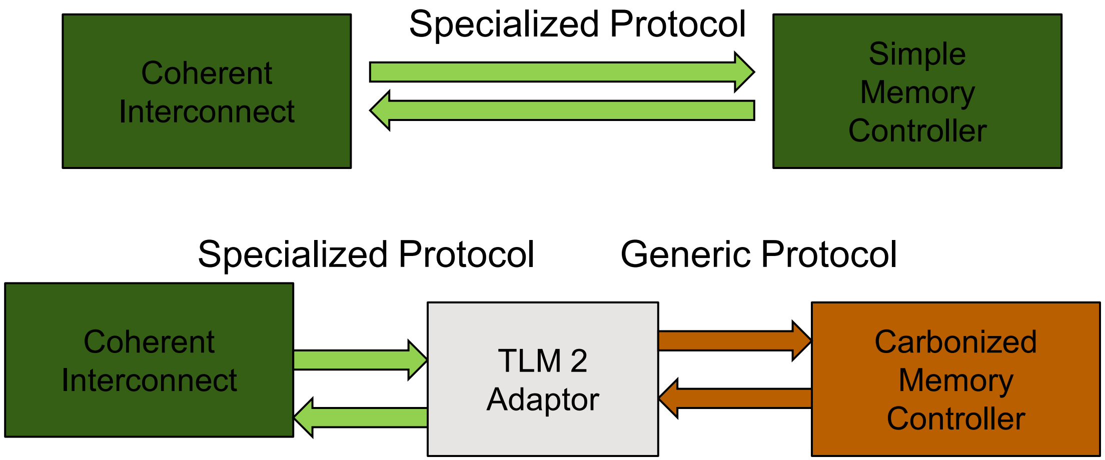
AXM5500 Model Memory Controllers

Two memory controller models were used:

- A simple model with fixed latencies
  - This was a SystemC simple model with parameterized fixed latency
  - Was used for debugging and quick and dirty analysis
- A cycle accurate model
  - A carbonized model with systemC wrapper
  - Was used it when detailed analysis of memory controllers were needed
  - This IP needed run-time license

AXM5500 Model Protocols and Sockets

- Connecting to third-party sockets can be challenging.
- Performance model IPs generally use generic protocol, non-blocking transport, and four-phase generic protocols.
- Some IPs used specialized protocols
- We used adaptors wherever needed. Example: memory controller connections



Summary

- We developed an architecture performance model for the AXM55XX SoC and relevant used case models for architecture validation.
- Some of the IPs used in the model were developed in-house and some came to us from third-party companies
- Third-party IP models were delivered to us as shared objects with the SystemC/TLM 2.0 interface.
- We used both generic and specialized payloads, non-blocking transport, and four-phase handshaking for socket connections and binding.
- We faced technical and legal challenges when integrating third-party IPs in our simulation environment.
- These challenges included technical interfacing issues, legal paperwork delays, importing these IPs in 3rd party architecture tools, and cost