2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION JAPAN

Data integrity checker for Coherency verification

Priyanshu Somvanshi, Shubhanshu Jain, Vaibhav Ashtikar.

Google

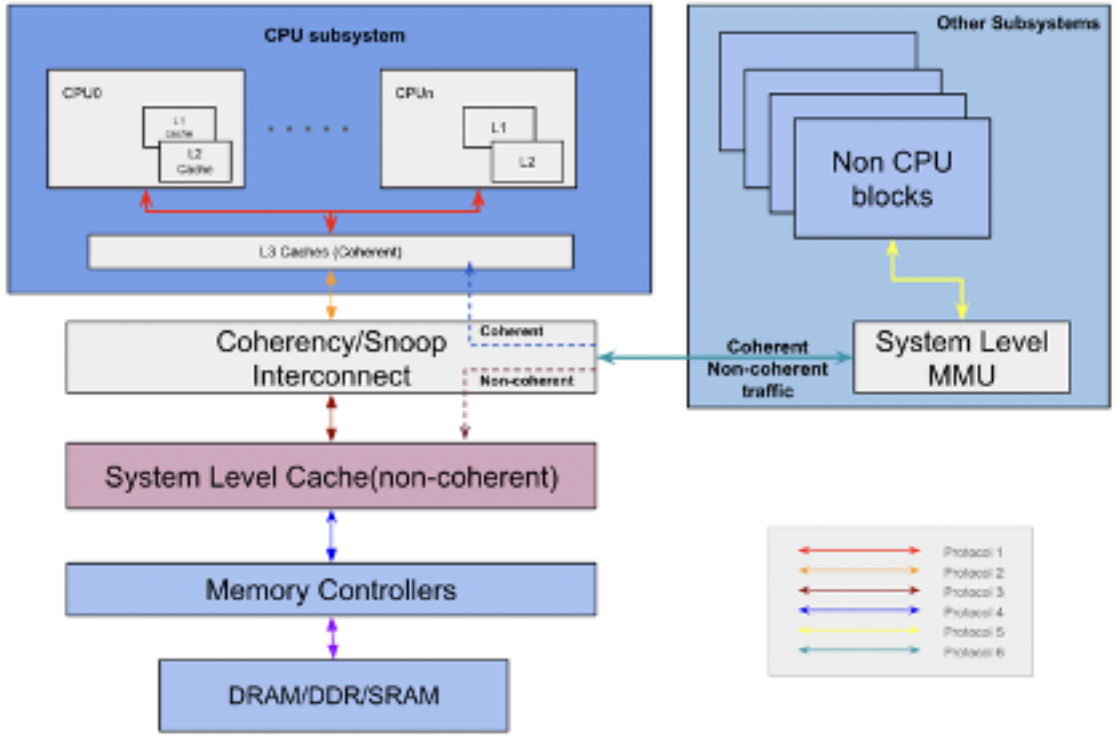
accellera SYSTEMS INITIATIVE

Abstract

- Coherency verification is a crucial aspect of chip verification.
- Variety of transaction protocols and protocol conversions across multiple interfaces presents new challenges for data integrity checks.
- Proposing Data integrity verification using interface monitors (trackers) and a post-processing tool.
- The tool aids in debugging data integrity issues by pinpointing the exact time when inconsistencies occurred.
- Post-processing reduces the simulation time consumed for verification runs and facilitates statistical analysis in regressions.

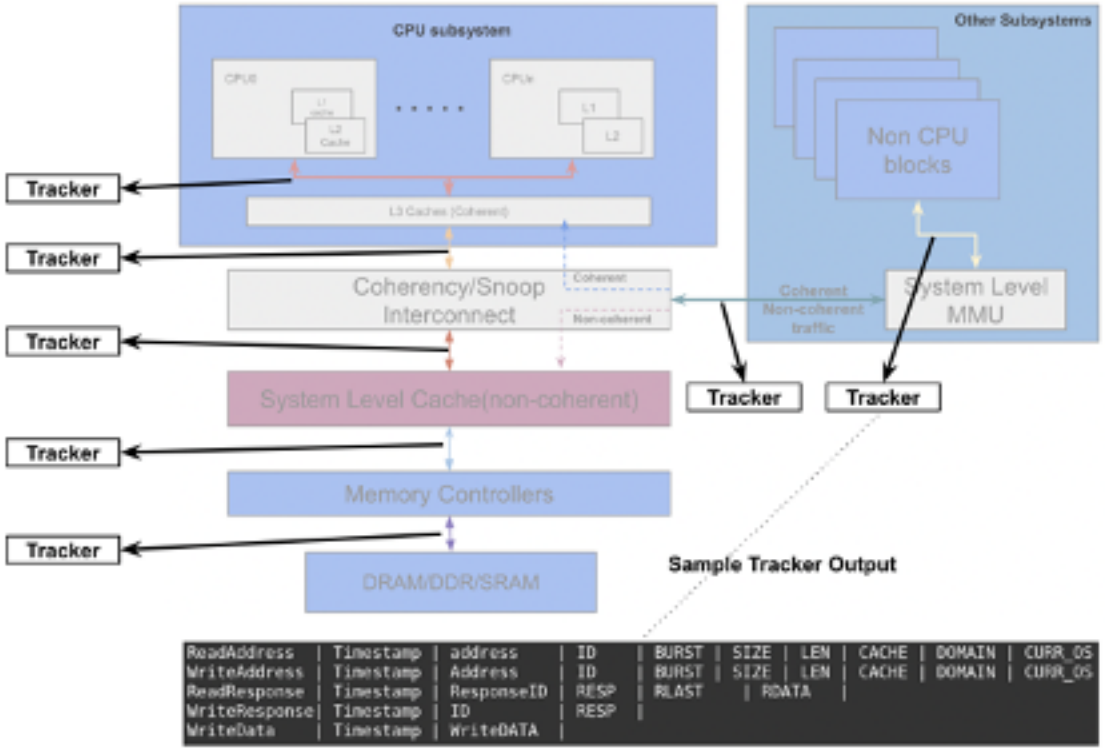
Introduction

- Coherent systems have
  - Caches(L1,L2, L3,SLC), interconnects, MC, SMMU,Memory
  - Various protocols governing transactions across interfaces
- Coherent Interconnect will route traffic using mem attr
  - Memory attributes will be set using MMU, SMMU
  - Interconnect uses mem attr for snoop/non-snoop traffic



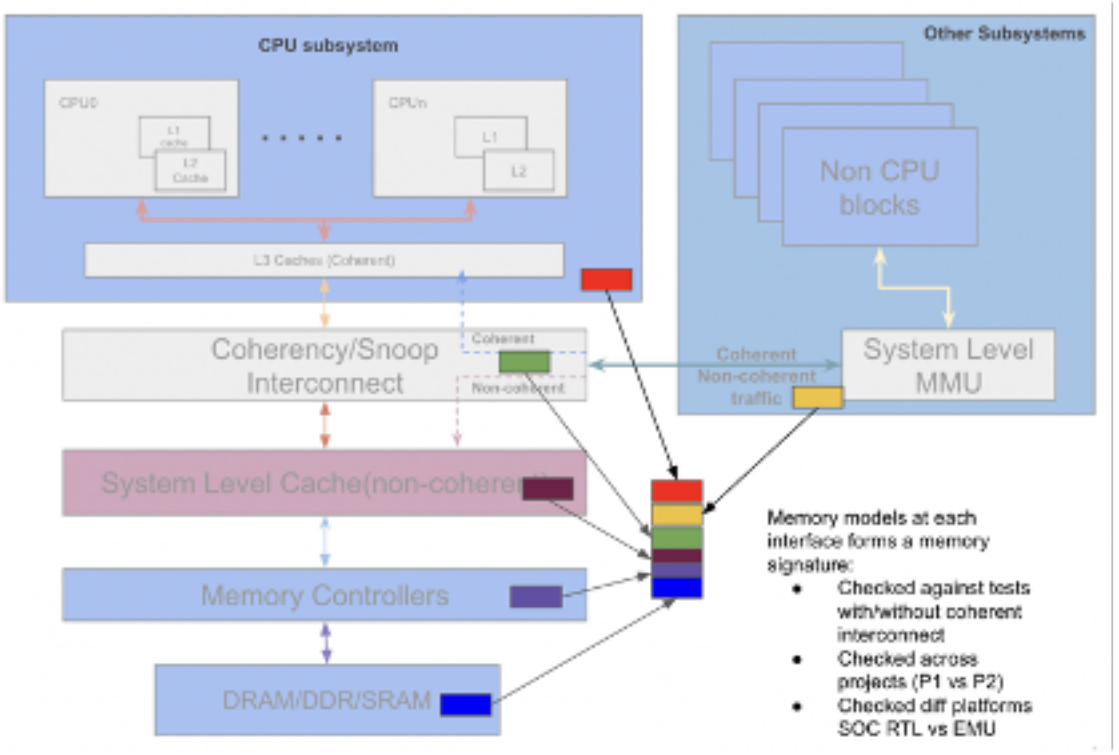
Methodology

- Trackers are installed at each interface
- Tracker outputs are serialised according to timestamp
- Transaction order is established
- Memory models are built across interfaces
- Each transaction response is validated with latest timestamp data in all Memory models



Memory Signature

- Memory models built at each interface contains latest timestamp data for certain addresses
- Memory models at all interface forms memory signature for all addresses at the end of simulation
- Signature can compared with
  - Testcase with/without coherent traffic
  - across different projects
  - RTL vs EMU runs



Results, Future Development

- Tool Successfully performed
  - Data Integrity check- data integrity check for every transaction
    - Tool pointed out exact start point for memory mismatches in coherency scenarios
    - Tool highlighted re-requests in coherent fabrics due to buffer saturations useful for initial configuration to be performed by SW
  - Functional coverage- functional coverage points were built using post-processing transactions
    - True sharing vs false sharing
    - Snoops responded using cache data or memory data ..?
- Tool can be enhanced for
  - Statistical Analysis- Tool if integrated with regression runs can provide overall statistical data e.g.
    - Performance- Data can be also used for performance calculations e.g. BW, latency
  - Architectural Rules verification-
    - AI/ML can be used for architecture rules construction using regression data