Solving Memory Subsystem Verification Challenges for Multi-Instance Designs

Shyam Sharma (Senior Software Architect, Cadence)

August 8, 2024

cadence

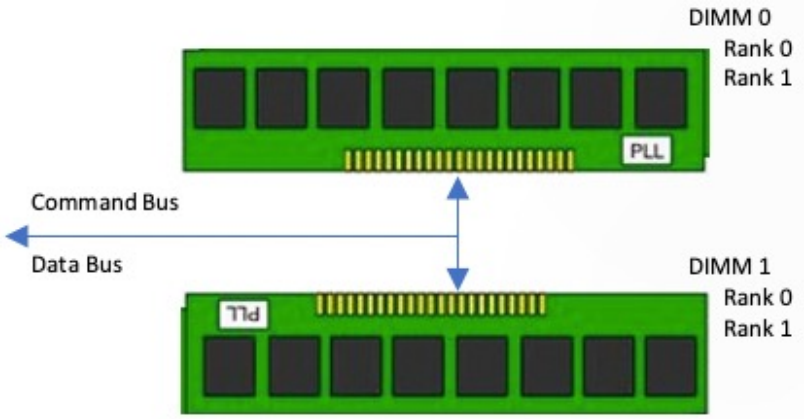
Agenda

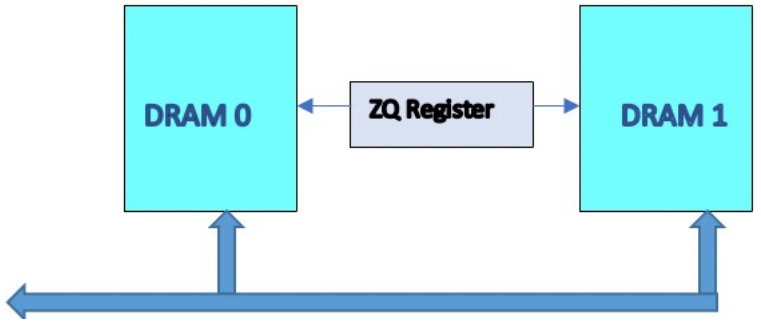
- Overview and importance of the inter-module checks
- Complexity of the inter-module checks
- Generic inter-connect solution for memory models
- Application, user examples
- Conclusion

Overview and Importance of the Inter-Module Checks

Multi-Component Subsystem: Background

- SoC/NoC/PCB have multiple components that
  - Share resources
  - Share I/O buses
  - Chronologically linked access order
- Sharing requires additional design constraints to
  - Avoid resource conflict
  - Prevent system instability
  - Prevent damage to the components
- Examples
  - Potential data clobbering/damage to the DRAMs for multiple data driver
  - ZQ register sharing
  - Target and non-target ODT settings

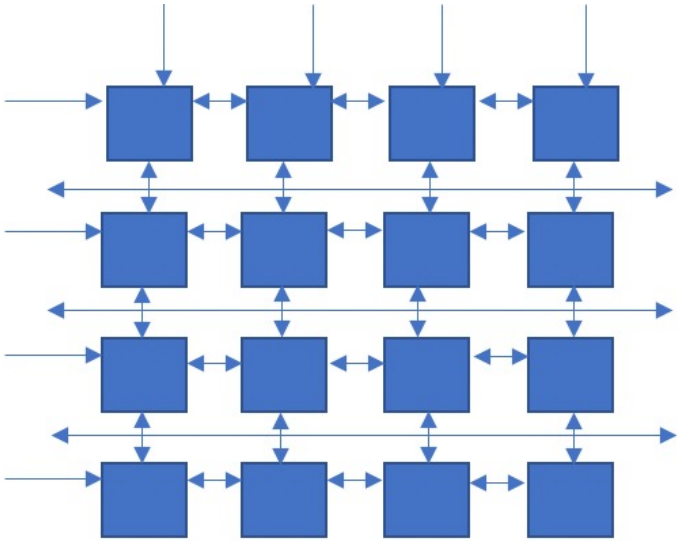




Complexity of the Inter-Module Checks and Innovative Generic Solution for Memory Models

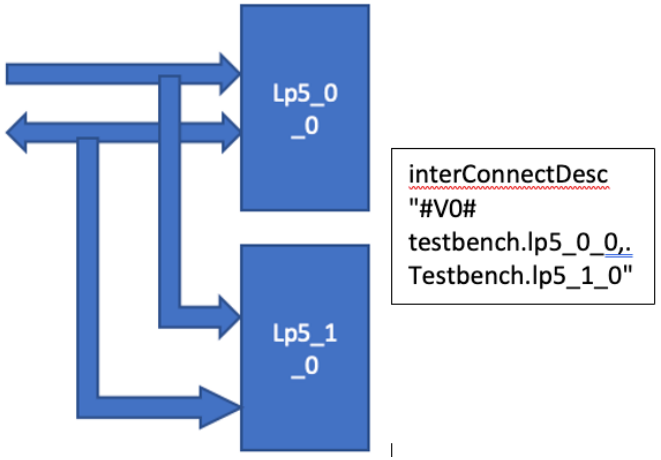
Complexity of the Inter-Module Checks for Memory Models

- Memory models are C/C++ Based Verification IP
  - Use VPI/DPI standard
  - No direct SystemVerilog design access
- Memory models are reactive
  - Receive signal values from its ports
  - Do checks for compliance of specification
- Memory models are traditionally limited to per instance in scope
- Real systems can use complex shared resources and I/O interconnects (figure)
- Memory model instance needs subsystem-level awareness



Generic Inter-Connect Solution for Memory Models

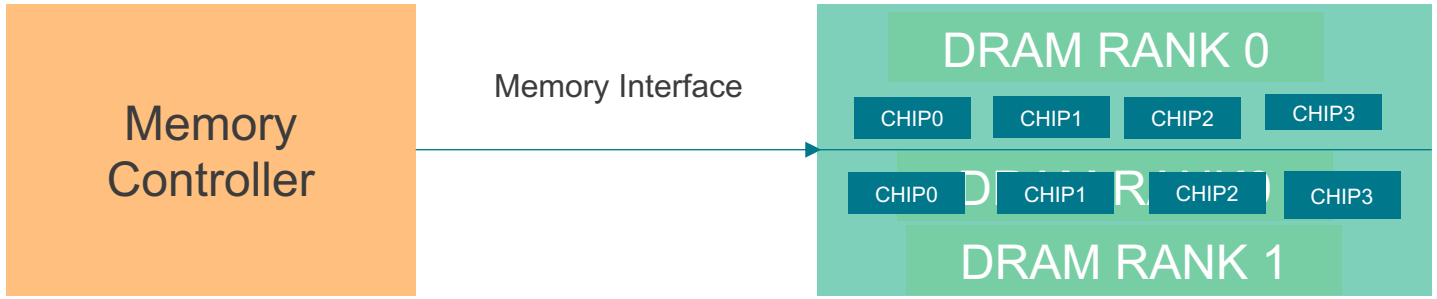
- Define new feature “interConnectDesc”
  - Define grammar to capture interconnects
- Create interconnect VIP module
  - Parse feature setting
  - Look up information of connected instances using VPI standard APIs
  - Internally create/associate instances with one another
- Implement hand-shaking between memory model instances
  - Pass events like commands
  - Share information on use of common resources
- Use triggers from another instance
  - Perform addition subsystem-level checks
- Modular, simple, and extendable
  - Can be extended to new VIP protocol family



Application and User Examples

User Examples

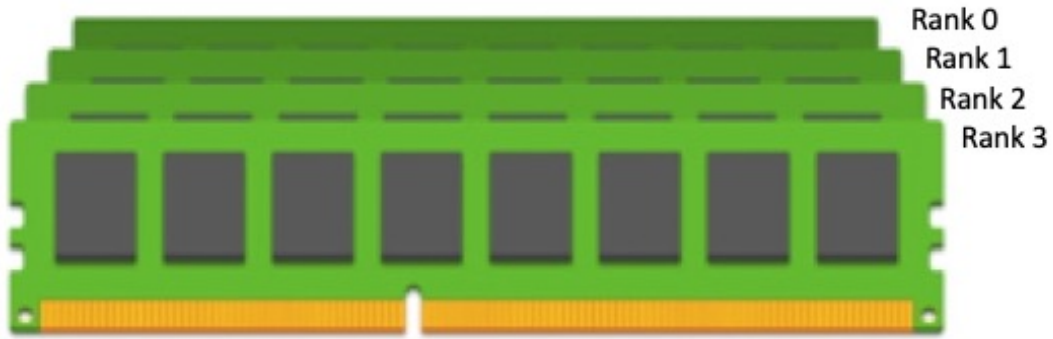
- The inter-rank timing checks is used to verify DDR controller timings where multiple memories are connected to one controller, also used actively by many IP/SoC customers of DDR VIP
- The inter-rank timing checks can be enabled through any instance of the DDR VIP



- If the instances are lpddr5_0_0 (rank 0 and component 0) and lpddr5_1_0 (rank 1 and component 0), you can issue mmsomaset as following:
  - `$mmsomaset("testbench.lp5_0_0", "interConnectDesc", "#V0#testbench.lp5_0_0,testbench.lp5_1_0","")`
- Can be extended to other DDRs like the one below shows DDR5 DIMM Rank 0 DRAM 0 shares the data bus with Rank 1 DRAM 0 and so on allowing to specify complex DIMM interconnect subsystem hierarchy :
  - `$mmsomaset("testbench.lp5_0_0", "interConnectDesc", #V0#uvm_test_top.ddr5dimmSve0.myUvmEnv.activeDevice.ddr5_0_0,uvm_test_top.ddr5dimmSve0.myUvmEnv.activeDevice.ddr5_1_0;…ddr5dimmSve0.myUvmEnv.activeDevice.ddr5_0_19,uvm_test_top.ddr5dimmSve0.myUvmEnv.activeDevice.ddr5_1_19”`

Application and User Examples

- There are Rank-to-Rank Command Timing Constraints that need to be met in order to be compatible/compliant to JEDEC specifications
  - Timing between **read** from one rank to **read** from other rank
  - Timing between **read** from one rank to **write** from other rank
  - Timing between **write** from one rank to **write** from other rank
  - Timing between **write** from one rank to **read** from other rank



- Example violation reported by model

```
M Denali Error: Detected[testbench.i0] RANK2RANK_READ_TO_READ_TIMING_VIOLATED
@686227652460 fs :: Rank to Rank read timing of 7 cycles is violated. Read was issued
before (BL/n_min + RU(tWCK2DQO_ranktorank/tCK) + RU((tRPRE + tRPST - 0.5tWCK)/tCK)) of
rank 1 expired. Read to rank1 was issued at 686026000500 fs.
```

- Rank-to-Rank timing violation errors

```
DENALI_LPDDR5_RANK2RANK_READ_TO_READ_TIMING_VIOLATED,
DENALI_LPDDR5_RANK2RANK_WRITE_TO_READ_TIMING_VIOLATED,
DENALI_LPDDR5_RANK2RANK_READ_TO_WRITE_TIMING_VIOLATED,
DENALI_LPDDR5_RANK2RANK_WRITE_TO_WRITE_TIMING_VIOLATED
```

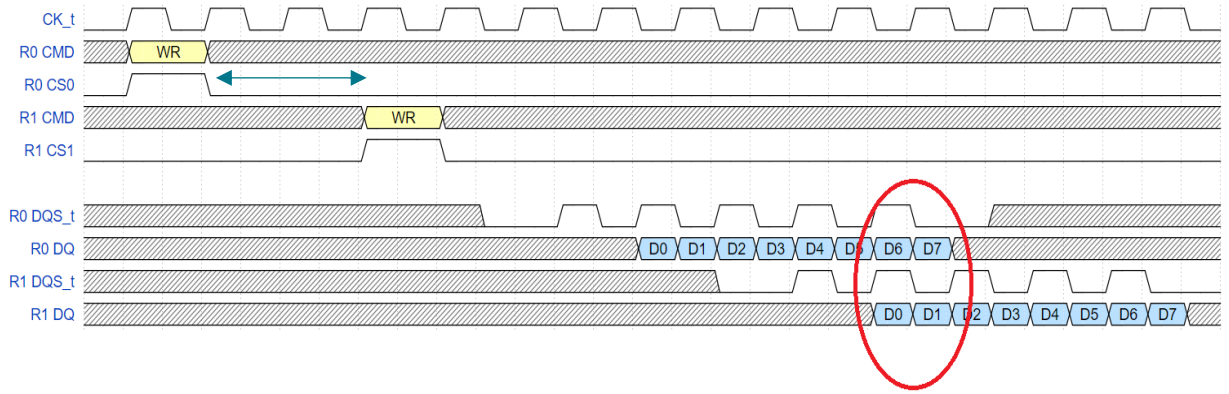
Application and User Examples

- Example showing timing violation between write from one rank to write from other rank:

- Timing violation if timing parameter between

  R0 WR and R1 WR is 3 clock cycles but

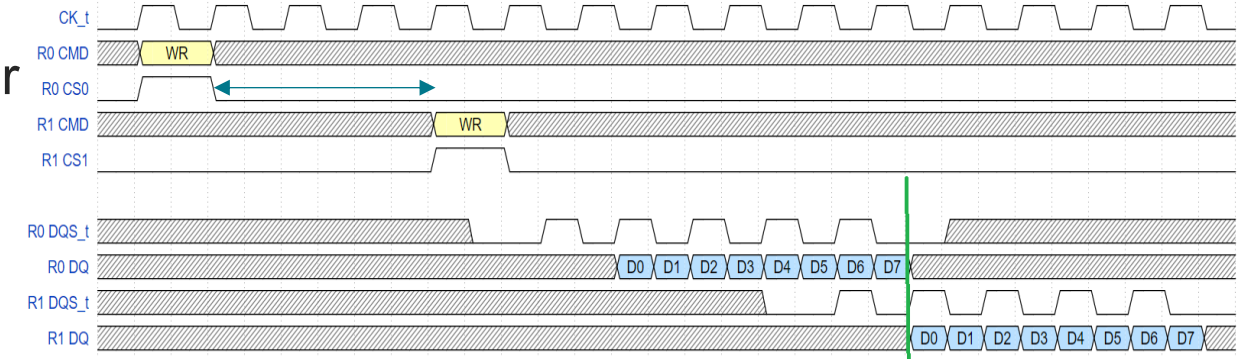
  R1 WR comes after 2 clock cycles



Data Overlap

- This is correct behavior where R1 WR is after

  3 clock cycles from R0 WR



No Data Overlap

Conclusion

Summary

- Simple modular and generic solution to define the subsystem hierarchy
- Can be extender to any homogeneous and heterogenous type of memory Verification IP
- Allows modeling and verification of any type of subsystem -or even system-level constraints
- Extensively used by both internal and external users of Cadence memory VIP with many glowing testimonies

Future Challenges

- Handle inter-module constraints for heterogeneous memory models
- Use shared memory between memory model instances to pass information efficiently
- Extend protocol transaction callbacks to include information from other memory instances