



Transaction-Level Design and Verification of a Hybrid AMBA AXI4-Lite/ACE Interconnect for Scalable Heterogeneous SoCs

T. Leelavathi
Assistant Professor
Leelareddy7791@gmail.com

Abdul Maqseed Shaik
Associate Professor
abdul.maqseed@gmail.com

Manga Rajalaxmi
rajalaxmimanga5@gmail.com

Cheguri Mahesh
maheshcheguri100@gmail.com

Dept. of ECE
Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus
Hyderabad, India

Corresponding Author: T. Leelavathi

DOI Link: <https://doi.org/10.64971/j.cph.eijtem.v13.i3.14.2026>

Abstract

Modern heterogeneous systems-on-chip require both lightweight register access and coherent shared-memory communication, yet these needs are often discussed as if AXI4-Lite and ACE were a single protocol layer. This paper presents a transaction-level architecture that deliberately separates an AXI4-Lite control plane from an ACE/ACE-Lite coherent data plane while sharing address-routing, arbitration, response-handling, monitoring, and verification services. The architecture targets CPU cores, accelerators, DMA engines, memory controllers, and memory-mapped peripherals. A deterministic cycle-abstract reference model is used to evaluate class-aware round-robin arbitration, filtered snooping, error responses, back-pressure, and cache-visible data consistency before RTL implementation. Verification covers 100,000 randomized address/response transactions, 20,000 cycles of continuous four-master contention, and 50,000 coherent memory operations, with no decode, response, mutual-exclusion, or stale-read violations in the reference model. At an offered load of 0.12 transactions/cycle, the proposed split-plane architecture reduces mean latency by 40.6%, 50.7%, and 76.3% for control-dominant, balanced, and memory-dominant mixes, respectively, compared with a serialized fixed-priority/broadcast-snoop baseline. Filtered snooping reduces peer snoop probes by 52.2-57.7%. A balanced-load sweep further shows that the baseline saturates near 0.154 transactions/cycle, whereas the proposed model sustains approximately 0.178 transactions/cycle at an offered load of 0.18. The results establish a reproducible architectural baseline for subsequent System Verilog/UVM implementation and FPGA/ASIC evaluation.

Keywords: AMBA, AXI4-Lite, ACE, ACE-Lite, system-on-chip, cache coherency, transaction-level modeling, UVM, interconnect, arbitration.

I. INTRODUCTION

The interconnect has become a first-order design constraint in heterogeneous SoCs. A contemporary device can contain general-purpose cores, hardware accelerators, DMA engines, shared caches, memory controllers, security blocks, and a large set of memory-mapped peripherals. These blocks do not have identical communication

requirements. Configuration and status registers favor short, low-complexity transfers, while shared-memory traffic must preserve data visibility across cached processors and non-cached I/O agents. The AMBA family addresses these requirements through related but distinct protocols: AXI4-Lite is optimized for simple register-style access, whereas ACE and ACE-Lite add coherency semantics to AXI-based system communication [1], [2].

The source design motivating this work identifies the correct high-level building blocks—master and slave interfaces, address decoding, arbitration, read/write control, snoop control, coherency management, response handling, peripheral connectivity, and transaction-based verification—but it treats the phrase “AXI4-Lite ACE interconnect” too broadly. For a publication-quality architecture, the protocol boundary must be explicit: ACE is not a coherency extension to AXI4-Lite. In the proposed model, AXI4-Lite therefore forms a control plane for single-beat peripheral register accesses, while ACE/ACE-Lite semantics are confined to the coherent memory plane. The planes share fabric-level services but not their protocol-specific signal semantics.

This separation yields two benefits. First, control traffic is prevented from unnecessarily inheriting the complexity of coherent transactions. Second, coherent memory traffic no longer competes with every low-bandwidth peripheral access in a single serialized path. The architecture can therefore exploit concurrency between independent traffic classes while preserving a common memory map, response model, observability framework, and verification flow. This is particularly relevant to accelerator-rich SoCs in which software frequently programs control registers while CPUs and DMA engines concurrently exchange data through shared memory.

Transaction-level modeling (TLM) is well suited to early exploration of such architectures because communication can be expressed as read/write operations with address, data, attributes, timing, and response rather than as every individual RTL transition. The TLM-2.0 standard explicitly supports memory-mapped bus modeling and architectural analysis [3]. In verification, the same transaction abstraction naturally maps to reusable sequence items, monitors, scoreboards, protocol checkers, and coverage models, consistent with the reuse objectives of UVM [4].

This paper makes four contributions. First, it formalizes a hybrid AXI4-Lite/ACE organization that keeps the control and coherent protocol domains distinct. Second, it introduces a class-aware round-robin arbitration model and a directory-assisted snoop filter to reduce interference and unnecessary coherent probes. Third, it defines a transaction-level verification plan covering routing, response classification, back-pressure, arbitration fairness, transaction conservation, and coherent visibility. Fourth, it reports reproducible quantitative results from a cycle-abstract reference model, including latency, throughput, fairness, and snoop-traffic metrics. The reported data are architectural simulation results; no RTL area, frequency, or power claims are made.

II. RELATED WORK AND RESEARCH GAP

AXI4-Lite remains widely used for software-visible control paths because it retains AXI’s independent read and write channels while removing burst complexity. Recent verification studies have focused on AXI4-Lite arbitration, read/write correctness, and UVM-based reuse. Babu et al. compared System Verilog and UVM-based verification for an AXI4-Lite arbitration design and emphasized UVM’s scalability for complex verification environments [8]. Sangani and Mehta similarly demonstrated UVM-based verification of AXI4-Lite read and write transactions [9]. These works are useful for control-plane correctness but do not address cache-coherent data sharing.

Coherency introduces a different class of correctness obligations. Kriouile and Serwe developed a formal model of an ACE-compliant coherent SoC and showed that cache-coherent interconnect verification must account for system-level state interactions, not merely local read/write behavior [6]. Their later work demonstrated how a formal model can improve the verification of a cache-coherent SoC [7]. These studies motivate the explicit coherency checks used here, although the present work uses a cycle-abstract executable reference model rather than exhaustive model checking.

The 2017 work by Kumar and Gurunadha Babu proposed an “AMBA AXI4-Lite ACE” interconnect concept using transaction-based design techniques [5]. It provided a useful integration direction but offered limited quantitative performance evidence and did not sharply separate AXI4-Lite control transfers from ACE coherent transactions. The present paper retains the useful modular structure of that work while correcting the protocol-domain interpretation and adding a reproducible evaluation methodology.

More recent research confirms that arbitration, interference, and transaction observability remain important in high-performance AXI systems. AXI-REALM adds modular traffic regulation and monitoring to AXI4 interconnects and demonstrates that interconnect-level control can materially improve predictability under contention [11]. Liang et al. propose transaction monitoring for reliable AXI4 systems, including timing and protocol-failure observability [12]. Wang et al. show that bus and memory collisions can significantly influence heterogeneous-SoC performance estimates [13]. These studies reinforce the need to treat the interconnect as an architectural resource rather than as passive wiring.

The research gap addressed here is therefore not a new wire-level AMBA protocol. It is the architecture and verification of a mixed-requirement fabric in which simple control traffic and coherent memory traffic coexist, remain semantically correct, and can be evaluated before RTL. The proposed contribution is intentionally positioned at the transaction-architecture level: it defines how requests are classified, routed, arbitrated, checked, and measured, then supplies reference results that can later be used as a golden target for SystemVerilog/UVM and FPGA implementations.

III. PROPOSED HYBRID INTERCONNECT ARCHITECTURE

The proposed organization separates software-visible control traffic from coherent shared-memory traffic while retaining common routing, arbitration, response, monitoring, and verification services. CPU cores are treated as fully coherent ACE-class agents. DMA and accelerator interfaces use ACE-Lite/I/O-coherent semantics when they require coherent visibility without maintaining a snooperable local cache. Low-bandwidth GPIO, UART, SPI/I2C, timer, and configuration registers remain on the AXI4-Lite control plane.

A transaction is represented by the tuple $T = \langle m, op, a, d, attr, r \rangle$, where m is the initiating master, op denotes read or write, a is the address, d is the write data or returned read data, $attr$ contains protocol-class and coherency attributes, and r is the response. The first architectural decision is the classification function $C(T)$: peripheral-register addresses enter the AXI4-Lite plane, shared cacheable memory enters the coherent plane, and unmapped addresses terminate at the response handler with DECERR.

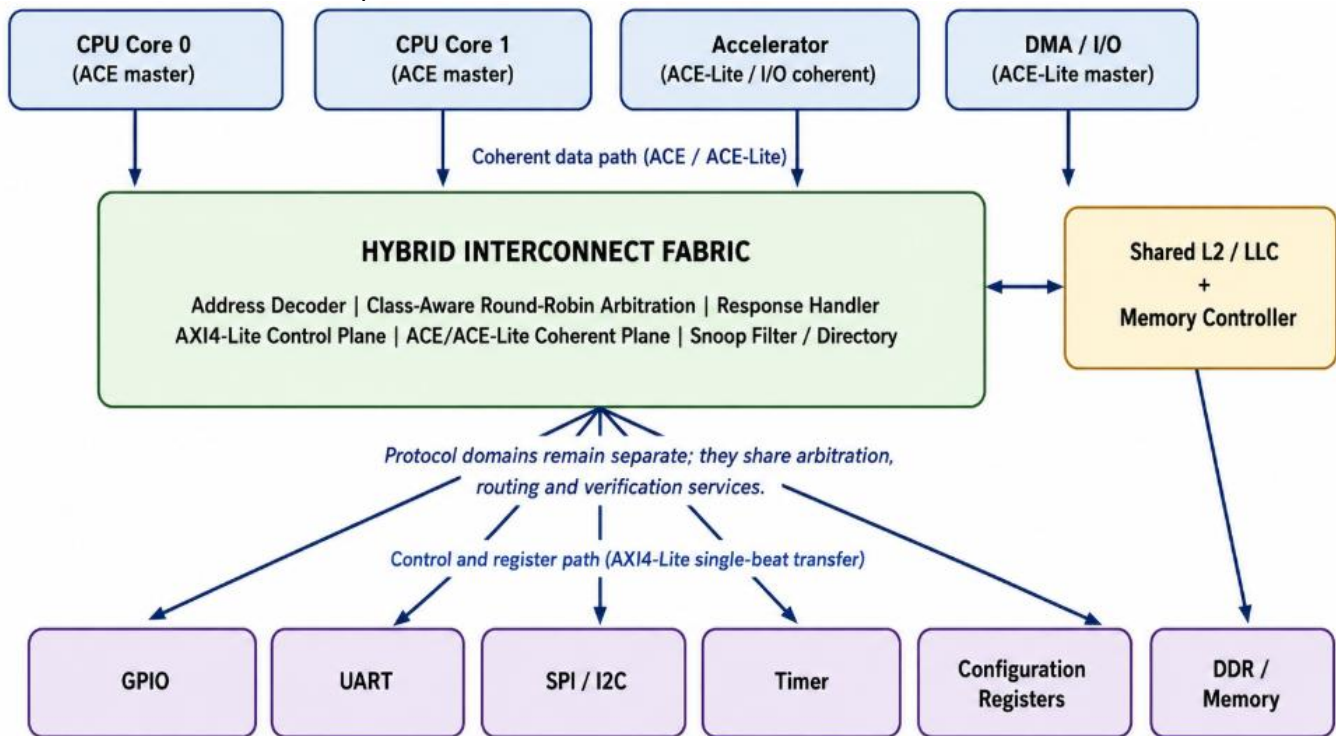


Fig. 1. Proposed hybrid AXI4-Lite control plane and ACE/ACE-Lite coherent data plane.

Figure 1 shows how the two protocol domains share fabric-level services without sharing protocol-specific signal semantics. A shared L2/last-level cache and memory controller terminate the coherent data plane, while memory-mapped peripherals terminate the control plane. The central fabric performs address classification, arbitration, response generation, transaction monitoring, and performance accounting.

The control plane uses single-beat transactions. At signal level, AXI4-Lite contains independent AW, W, B, AR, and R channels; a transfer on a channel completes only when VALID and READY are simultaneously asserted [1]. The reference model abstracts this handshake as a service interval with optional back-pressure. It nevertheless checks the transaction-level invariant that an accepted request is neither dropped nor duplicated and that exactly one terminal response is produced.

The coherent plane models two forms of agents. ACE-class masters can own or share cached copies and therefore participate in snoop-based visibility actions. ACE-Lite-class I/O agents do not expose a snooperable local cache but require coherent access to system memory. Rather than broadcasting every coherent request to all peers, the proposed model maintains a lightweight directory of potential cache holders. A snoop is generated only toward a

master that may contain the requested line. This abstraction is not a replacement for the full ACE state machine; it is an architectural mechanism for estimating the traffic benefit of filtered rather than unconditional peer probes.

Table I. Memory map used by the transaction-level model.

Address range	Target	Traffic class
0x00000000 - 0x00000FFF	GPIO	AXI4-Lite control
0x00001000 - 0x00001FFF	UART	AXI4-Lite control
0x00002000 - 0x00002FFF	SPI/I2C	AXI4-Lite control
0x00003000 - 0x00003FFF	Timer	AXI4-Lite control
0x00004000 - 0x00004FFF	Configuration	AXI4-Lite control
0x10000000 - 0x1FFFFFFF	Memory	ACE/ACE-Lite coherent

Arbitration and response handling. Each plane uses a rotating round-robin pointer so that a continuously requesting master is served at least once every N grants for N active masters. The control and coherent planes may operate concurrently because they terminate at different classes of resources. Within a shared target, only one winner is admitted at a time. Successful accesses return OKAY; unmapped addresses return DECERR; injected target failures return SLVERR. The response handler also records latency and transaction-class statistics for post-simulation analysis.

Performance measures. For transaction i , latency is $L_i = t_{resp,i} - t_{req,i}$. Aggregate throughput is $X = N_{complete} / T_{sim}$. Arbitration fairness is summarized by Jain's index $J = (\sum x_m)^2 / (M \sum x_m^2)$, where x_m is the number of completed transactions from master m . Snoop reduction is $R_s = 1 - N_{filtered} / N_{broadcast}$, where $N_{broadcast}$ is the number of peer probes that an all-peer broadcast policy would issue for the same coherent accesses.

IV. TRANSACTION-LEVEL MODELING AND VERIFICATION METHODOLOGY

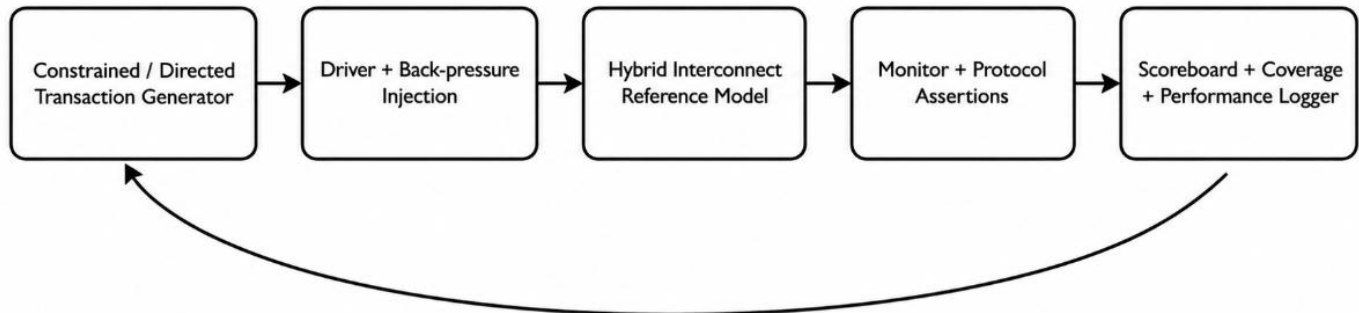


Fig. 2. Transaction-level verification and performance-measurement flow.

The executable model is deterministic for a fixed seed and advances in discrete cycles. Four masters generate independent transactions. Control requests have short service times (2-4 cycles), while coherent memory requests use 5-9 base cycles plus a small snoop penalty when another cache may hold the line. The model is intentionally approximately timed rather than cycle-accurate RTL; its purpose is architectural comparison and early verification. The same transaction fields can later be mapped to SystemC TLM payload extensions or SystemVerilog UVM sequence items.

Two architectures are evaluated. The baseline serializes all traffic through one fixed-priority service point and models coherent access using broadcast snooping to all three peer masters. The proposed architecture uses separate control and coherent service planes, round-robin arbitration within each plane, and directory-filtered snooping. The baseline is not intended to represent a particular commercial interconnect; it is a deliberately simple comparison point that exposes the cost of combining unrelated traffic and of unconditional peer snoops.

The verification flow in Fig. 2 combines directed scenarios with randomized transactions. The generator creates mapped and unmapped addresses, reads, writes, injected target errors, simultaneous requests, and coherent read/write sequences. Monitors reconstruct completed transactions, while a scoreboard checks address-to-target mapping and response classification. Arbitration assertions enforce one-hot grant behavior and bound the grant gap under continuous contention. A small versioned-memory model verifies that a coherent read never observes a version older than the most recent committed write.

The methodology also distinguishes functional verification from performance experiments. Functional tests are pass/fail checks and are not used to tune the architecture. Performance experiments use a stable random seed, fixed service-time distributions, and a warm-up removal of the first 10% of completed transactions. The offered load is expressed as aggregate generated transactions per cycle across the four masters. Three workload mixes are

evaluated at 0.12 transactions/cycle: 70/30 control/coherent, 50/50, and 30/70. A separate balanced-load sweep varies the offered load from 0.04 to 0.18 transactions/cycle.

Table II. Reference-model parameters for reproducible evaluation.

Parameter	Value
Masters	4 (2 CPU-class, 1 accelerator-class, 1 DMA/I/O-class)
Mapped targets	GPIO, UART, SPI/I2C, Timer, Configuration, Memory
Control service time	2-4 cycles
Coherent base service time	5-9 cycles
Write fraction in coherent traffic	35%
Shared-line locality bias	35% reuse opportunity
Arbitration	Baseline: fixed priority; Proposed: round-robin per plane
Coherency traffic	Baseline: broadcast; Proposed: directory-filtered peer probes
Scenario experiment	250,000 cycles, seed 2026, offered load 0.12
Load sweep	120,000 cycles/point, seed 777, balanced 50/50 traffic

V. RESULTS AND DISCUSSION

A. Functional verification. The reference model was first exercised with randomized address/response transactions, continuous-contention arbitration, and coherent memory operations. Table III reports the checks. All 100,000 address transactions were decoded to the expected target or to the unmapped region; the same set produced the expected OKAY, SLVERR, or DECERR response classification. Under 20,000 cycles in which all four masters continuously requested service, the round-robin arbiter issued exactly 5,000 grants to each master, maintained one-hot grant behavior, and bounded the inter-grant distance to four cycles. The coherence test executed 50,000 read/write operations over 256 cache-line indices and observed no stale-read violation in the versioned-memory model.

Table III. Transaction-level verification results.

Check	Stimulus	Observed result	Status
Address decode	100,000 randomized mapped/unmapped accesses	92,043 mapped; 7,957 unmapped; 0 misroutes	PASS
Response classification	100,000 transactions including 1,871 injected target errors	0 response mismatches	PASS
Arbitration mutual exclusion	20,000 continuous-contention cycles	0 multi-grant violations	PASS
Round-robin bounded service	4 continuously requesting masters	Grant counts [5000, 5000, 5000, 5000]; max gap 4 cycles	PASS
Coherent visibility	50,000 versioned memory operations	0 stale reads; 25,716 invalidations	PASS

B. Workload-level performance. Table IV compares the serialized baseline and the proposed split-plane fabric at an aggregate offered load of 0.12 transactions/cycle. The strongest benefit appears when coherent traffic dominates because the baseline combines longer coherent service times, broadcast snoop overhead, and fixed-priority contention in one queue. The proposed design lowers mean latency by 40.6% in the control-dominant case, 50.7% in the balanced case, and 76.3% in the memory-dominant case. The 95th-percentile reduction is equally important: in the memory-dominant workload the P95 latency falls from 501 to 74 cycles. Throughput remains approximately equal to the offered load in these three experiments, indicating that the main gain at this operating point is reduced queueing rather than a change in generated work.

Table IV. Performance at 0.12 transactions/cycle aggregate offered load.

Workload	Base avg L	Prop. avg L	Reduction	Base P95	Prop. P95	Snoop reduction
Control-dominant (70/30)	11.81	7.02	40.6%	32	16	57.7%

Balanced (50/50)	21.53	10.62	50.7%	73	29	54.2%
Memory-dominant (30/70)	95.20	22.57	76.3%	501	74	52.2%

The snoop-filter results are significant even in this simplified model. For the three workload mixes, peer-probe traffic is reduced by 52.2-57.7% relative to probing every other master on each coherent access. The exact reduction depends on sharing locality: higher reuse creates more legitimate directory hits and therefore more necessary probes. The result should not be interpreted as an ACE protocol-compression claim; it measures only the number of peer checks avoided by a directory that can rule out caches that do not contain the line.

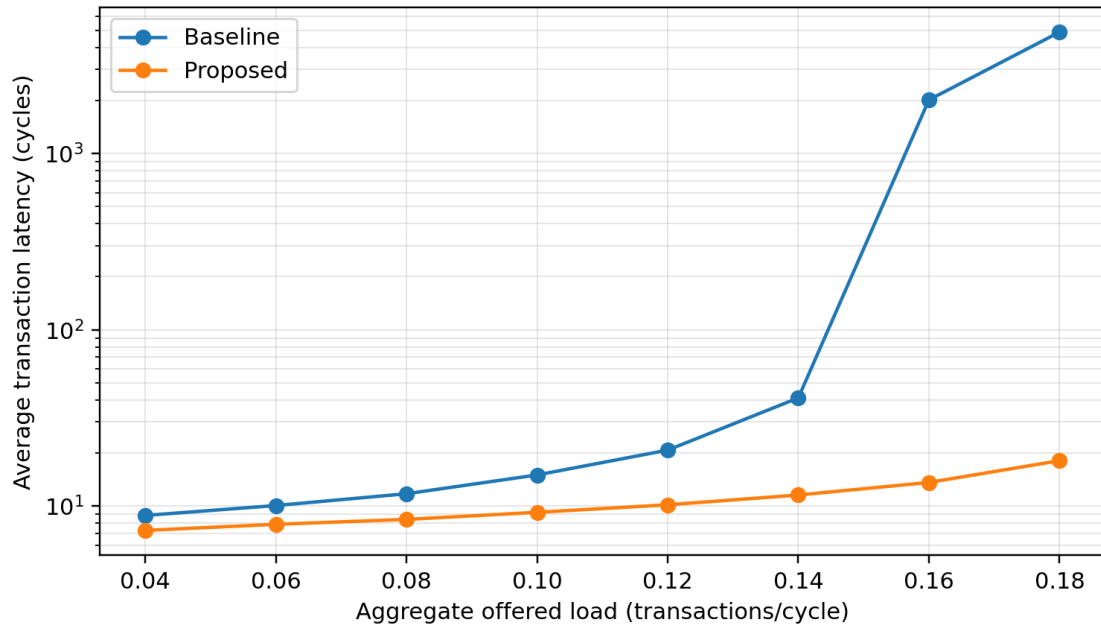


Fig. 3. Average transaction latency versus balanced offered load (logarithmic y-axis).

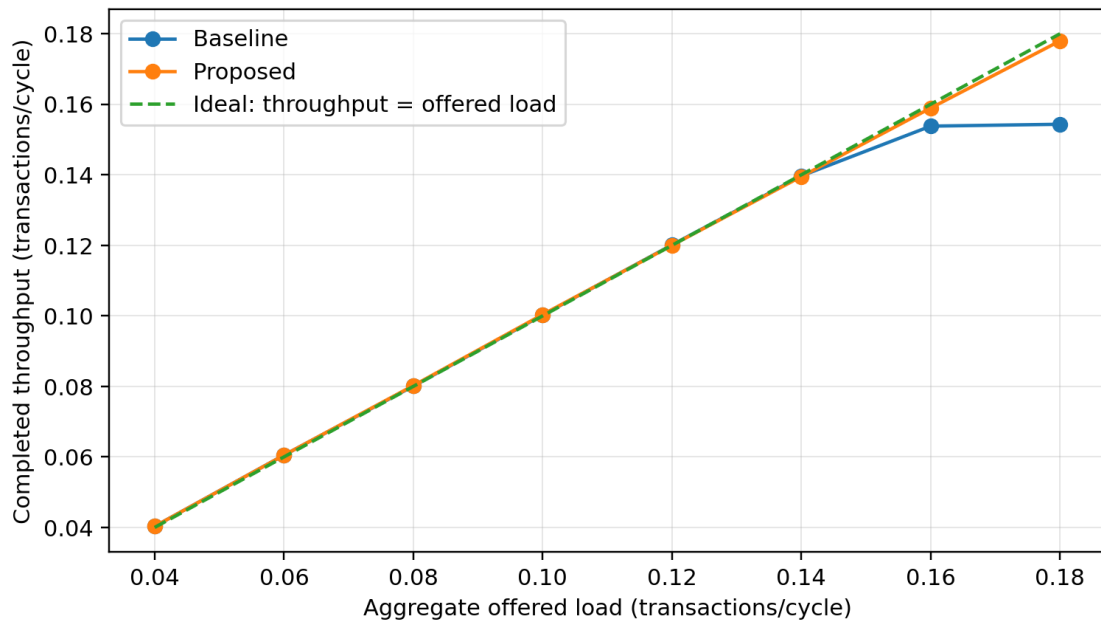


Fig. 4. Completed throughput versus balanced offered load.

Figures 3 and 4 show the balanced 50/50 load sweep. At low load, both designs track the offered traffic rate, but the proposed architecture consistently has shorter latency because control transactions can proceed independently

of coherent memory service. The divergence becomes pronounced as the offered load approaches the service limit of the serialized baseline. At 0.14 transactions/cycle, the baseline already exhibits a long P95 tail (183 cycles in the sweep), while the proposed architecture keeps P95 latency at 34 cycles. At offered load 0.16, the baseline completes only about 0.154 transactions/cycle and its queue grows rapidly, whereas the proposed design completes about 0.159 transactions/cycle with a P95 of 44 cycles.

At the highest evaluated load of 0.18 transactions/cycle, the serialized baseline remains saturated near 0.154 transactions/cycle. The proposed model sustains about 0.178 transactions/cycle because the two planes can serve unrelated traffic concurrently. This is an architectural concurrency benefit rather than a claim that the proposed fabric can universally double performance: if nearly all traffic targets one coherent memory resource, the coherent plane itself becomes the bottleneck. The memory-dominant scenario in Table IV already shows this trend through a higher proposed latency than the other mixes.

Fairness is preserved by round-robin arbitration. Across the proposed workload experiments, Jain's fairness index is greater than 0.9998. More importantly, the directed continuous-contention test provides a deterministic starvation bound: with four permanently active masters, every master is granted once in each four-grant rotation. A production interconnect could replace this policy with weighted round-robin or QoS-aware regulation, but the current policy provides a transparent golden model for functional comparison.

The results also illustrate why control/coherent separation should be considered before RTL optimization. If one begins with a monolithic interconnect, later improvements such as deeper buffers or more aggressive arbitration must fight avoidable head-of-line blocking. In contrast, classifying traffic at the architecture boundary makes independent service and verification explicit. This can simplify future physical implementation because the AXI4-Lite control path can remain narrow and low-complexity while coherent data paths are provisioned according to memory bandwidth requirements.

Limitations are intentionally stated. The current model does not implement the complete ACE signal set, all legal cache states, barriers, ordering rules, multiple outstanding AXI IDs, burst transfers, clock-domain crossings, or physical timing. The latency numbers therefore represent the specified abstract service model, not silicon timing. Area, power, maximum frequency, and exact protocol coverage must be established in a subsequent RTL/UVM implementation. These limitations do not invalidate the architectural comparison because both baseline and proposed designs are evaluated under the same deterministic transaction model, but they define the scope of the claims.

VI. CONCLUSION AND FUTURE WORK

This paper reformulates the original AXI4-Lite/ACE interconnect idea into a technically explicit hybrid architecture for heterogeneous SoCs. AXI4-Lite is used only where its simplicity is advantageous—software-visible control and status registers—while ACE/ACE-Lite semantics are reserved for shared-memory traffic that requires coherent visibility. Address classification, arbitration, response handling, monitoring, and verification are shared at the fabric level without conflating the two protocol domains.

A transaction-level reference model was used to verify routing, response classes, arbitration, transaction conservation, and coherent data visibility, and to measure latency, throughput, fairness, and snoop traffic. The proposed split-plane architecture reduced mean latency by 40.6-76.3% across three traffic mixes at 0.12 transactions/cycle and reduced peer snoop probes by 52.2-57.7% relative to a serialized broadcast-snoop baseline. The balanced-load sweep further showed that the proposed model sustains higher offered load before saturation. The next step is a synthesizable SystemVerilog realization with separate AXI4-Lite and ACE/ACE-Lite interface adapters, a parameterized address decoder, round-robin/weighted arbitration, and an executable directory/snoop-filter block. UVM should then reuse the transaction definitions and scoreboards developed at the reference-model level. RTL evaluation should report functional coverage, assertion coverage, FPGA/ASIC resource use, maximum frequency, dynamic power, and protocol-compliance regressions. A further extension can replace the simple two-plane fabric with a scalable coherent NoC while preserving the same classification, observability, and verification principles.

REFERENCES

- [1] Arm Ltd., "AMBA AXI and ACE Protocol Specification," Arm IHI 0022E, 2013.
- [2] Arm Ltd., "AMBA 4: AXI4, AXI4-Lite, ACE and ACE-Lite," AMBA system architecture documentation, accessed Aug. 2026.
- [3] Open SystemC Initiative, "OSCI TLM-2.0 Language Reference Manual," 2009.
- [4] Accellera Systems Initiative, "Universal Verification Methodology (UVM) 2020 Reference Implementation for IEEE 1800.2," release 2020-3.1, 2024.
- [5] C. Kumar and M. Gurunadha Babu, "A Design of AMBA AXI4-Lite ACE Interconnect Protocol for Transaction-based SoC Design Techniques Integration," *International Journal of Engineering and Computer Science*, vol. 6, no. 6, pp. 21870-21874, 2017, doi: 10.18535/ijecs/v6i6.53.

- [6] A. Kriouile and W. Serwe, "Formal Analysis of the ACE Specification for Cache Coherent Systems-on-Chip," in Formal Methods for Industrial Critical Systems (FMICS 2013), LNCS 8187, pp. 108-122, 2013, doi: 10.1007/978-3-642-41010-9_8.
- [7] A. Kriouile and W. Serwe, "Using a Formal Model to Improve Verification of a Cache-Coherent System-on-Chip," in Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2015), LNCS 9035, pp. 708-722, 2015.
- [8] A. R. Babu, P. Anand, Y. Kim, and S. Jadhav, "System Verilog versus UVM-based Verification of AXI4-Lite Arbitration," in Proc. IEEE SoutheastCon, 2023, doi: 10.1109/SoutheastCon51012.2023.10115141.
- [9] H. Sangani and U. Mehta, "UVM based Verification of Read and Write Transactions in AXI4-Lite Protocol," in Proc. IEEE Region 10 Symposium (TENSYP), 2022, pp. 1-5, doi: 10.1109/TENSYP54529.2022.9864552.
- [10] A. Jain, H. Gupta, S. Jana, and K. Kumar, "Early Development of UVM based Verification Environment of Image Signal Processing Designs using TLM Reference Model of RTL," International Journal of Advanced Computer Science and Applications, vol. 5, no. 2, 2014, doi: 10.14569/IJACSA.2014.050212.
- [11] T. Benz, A. Ottaviano, R. Balas, A. Garofalo, F. Restuccia, A. Biondi, and L. Benini, "AXI-REALM: A Lightweight and Modular Interconnect Extension for Traffic Regulation and Monitoring of Heterogeneous Real-Time SoCs," in Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2024, pp. 1-6.
- [12] C. Liang, T. Benz, A. Ottaviano, A. Garofalo, L. Benini, and D. Rossi, "Towards Reliable Systems: A Scalable Approach to AXI4 Transaction Monitoring," in Proc. Design, Automation & Test in Europe Conference (DATE), 2025, doi: 10.23919/DATE64628.2025.10992911.
- [13] J. Wang, Y. Gim, S. Park, and C. S. Park, "Modeling and Simulation of System Bus and Memory Collisions in Heterogeneous SoCs," IEEE Access, vol. 10, pp. 25901-25921, 2022, doi: 10.1109/ACCESS.2022.3154014.

How do I cite this article?

T. Leelavathi et al, Transaction-Level Design And Verification Of A Hybrid Amba Axi4-Lite/Ace Interconnect For Scalable Heterogeneous Socs, Excel International Journal of Technology, Engineering and Management, 2026;Volume -13, Issue-3_Page_157-164. <https://doi.org/10.64971/j.cph.ejitem.v13.i3.14.2026>



This is an open access article under the CC BY-NC-ND license
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)