Multicore Cache Coherence: A UVM-Based Study of Quad-Core MESI Coherence

Muhammad Zeeshan Malik[1], Abeera Adnan[2], Ammar Saqib[3]
2021ee143@student.uet.edu.pk[1] 2021ee106@student.uet.edu.pk[2] 2021ee141@student.uet.edu.pk[3]
Department of Electrical Engineering University of Engineering and Technology, Lahore

Abstract

This paper presents the design and verification of a quad-core L1 cache coherence system utilizing the snooping- based MESI (Modified, Exclusive, Shared, Invalid) protocol. The design ensures cache coherence in a shared-memory multicore environment while addressing challenges such as transient states, race conditions, and arbitration fairness. Universal Verification Methodology (UVM) is employed to simulate and validate the system, ensuring correctness under various inter-core interaction scenarios. The verification process focuses on detecting coherence violations, ensuring bus arbitration fairness, and validating the functionality of individual components, including the L1 cache and arbiter. Simulation results demonstrate the system's ability to maintain coherence and fairness under diverse conditions, providing a robust framework for multi-core cache systems.

Key Words

Cache coherence, MESI Protocol, UVM, Quad-Core Systems, Snooping, Bus Arbitration, Multithreaded applications.

1. Introduction

The increasing demand for high-performance computing has led to the widespread adoption of multicore processors. However, maintaining cache coherence in such systems remains a critical challenge, especially when multiple cores share a common memory space. The MESI (Modified, Exclusive, Shared, Invalid) protocol is a widely used solution, but its hardware implementation introduces challenges such as race conditions, transient states, and bus contention. This paper presents the design and verification of a quad-core L1 cache coherence system using the snooping-based MESI protocol. The Universal Verification Methodology (UVM) is employed to simulate and validate the system, ensuring correctness and robustness under various conditions. The significant contributions of this manuscript are as follows:

1) A novel quad-core L1 cache architecture implementing the MESI protocol with an atomic snooping bus.
2) A UVM-based verification framework to validate cache coherence at both component and system levels.
3) An in-depth analysis of transient states, race conditions, and arbitration fairness in a multicore environment.
4) Extensive simulation-based evaluation to ensure correct- ness and efficiency under different workload conditions.

While the MESI protocol effectively addresses cache coherence at the architectural level, its hardware implementation introduces significant challenges. As processor designs grow increasingly complex with higher core counts, split-transaction buses, and aggressive performance optimizations, ensuring efficient and correct cache coherence becomes more difficult [1]. One major issue arises from the non-atomic nature of state transitions in hardware. In the architectural model, MESI state transitions are treated as atomic, representing a single indivisible step. However, in real hardware, each transition involves multiple instructions, and multiple cores may simultaneously attempt to access or modify the same cache line. This parallelism can lead to race conditions, where cores interfere with each other's operations, such as when two cores attempt to write to the same memory location concurrently [2]. Additionally, transient states further complicate implementation. During state transitions, such as from Shared to Exclusive, the system may enter intermediate states where it is neither fully in the old state nor the new state. These transient states must be carefully managed to prevent cores from reading or writing stale or invalid data, ensuring correctness and coherence across the system [3].

Modern processors employ split-transaction buses to enhance efficiency by allowing multiple transactions to proceed simultaneously. While this optimization improves performance, it significantly complicates the coherence protocol, as multiple memory operations can overlap without waiting for prior operations to be completed. This introduces additional transient states and potential timing issues, making the protocol harder to implement correctly [4]. Furthermore, verifying the correctness of a hardware implementation of the MESI protocol is inherently challenging. Simulation techniques,

while useful for testing specific scenarios, cannot exhaustively guarantee the absence of errors such as deadlock, livelock, or starvation across all possible cases [5]. The management of concurrent, non-atomic state transitions remains a critical challenge in multi-core systems, often leading to transient states, race conditions, and deadlock. These issues are exacerbated by the use of split-transaction buses, making it increasingly difficult to ensure correct, efficient, and scalable cache coherence across multiple cores [6]. As a result, robust verification methodologies, such as UVM, are essential to validate the correctness and reliability of MESI-based cache coherence systems in complex multi-core environments.

2. System Design and Implementation

The proposed quad-core system integrates private L1 caches for each core and a shared L2 cache to ensure coherence and efficient memory management. The MESI protocol gov- erns cache state transitions, while an arbiter and FIFO-based request buffering mechanism regulates memory requests and bus access. Each core generates memory requests that are directed to its dedicated L1 cache, where hits and misses are determined. Communication between cores and caches is facilitated through an atomic snooping bus, with an arbiter managing bus access to prevent conflicts. In the case of a cache miss, the request is forwarded to the L2 cache or main memory through the cache controller unit (CCU). A directly mapped cache replacement algorithm ensures efficient utilization of cache resources. The design focuses on the L1 cache, with interactions between L1 and L2 caches optimized for reliability and performance. The CCU enforces MESI coherence rules by monitoring bus transactions and coordinating memory updates. The arbiter ensures orderly access to the shared bus, resolving contention among cores and enforcing priority-based arbitration to prevent starvation.

Each core generates read and write requests, which are sent to the L1 cache controller along with the address (`pr_addr`) and data (`pr_data`) fields. If the L1 cache controller is occupied with a previous request, the core stalls until the controller becomes available. The L1 cache controller is responsible for processing incoming requests from its associated core, checking if the requested data is present, and updating the cache state accordingly. If the data is not available, it collaborates with the CCU to retrieve it from the L2 cache or main memory. The arbiter ensures that only one core or cache controller uses the bus at any given time, prioritizing requests to prevent race conditions and maintain atomicity. The FIFO buffer temporarily holds pending requests when the bus is busy, preventing stalls and optimizing memory access latency. It also decodes the `buf_out` signal to identify the requesting core before processing the request. The L2 cache controller plays a crucial role in managing shared data across cores, reducing direct accesses to main memory, and maintaining consistency by coordinating updates between L1 caches and main memory. The hierarchical memory structure—where L1 caches provide fast, low-latency access while L2 and main memory serve as shared resources—ensures efficient performance and balanced workload distribution across cores.

Main memory serves as the final level in the memory hierarchy; processing read and write requests when data is not found in the L1 or L2 caches. The L2 cache controller manages access to main memory, ensuring minimal access delays and maintaining memory consistency across all cores. Control signals such as `Bs_Grant` and `Snoop_Grant` are used to enforce cache coherence between cores, ensuring that data remains consistent and up to date. This structured approach ensures scalable and robust cache coherence management in a multi-core environment, addressing challenges related to concurrency, transient states, and arbitration fairness.

The system is configured with four CPU cores, each with a 4 KB private L1 cache and an 8 KB shared L2 cache. The MESI protocol is used to maintain cache coherence, with a snooping, priority-based bus facilitating communication between cores. This hierarchical design balances processing performance and memory access latency, with faster but smaller L1 caches and slower but larger L2 caches and main memory. The operational flow begins with a core checking its private L1 cache for requested data. If an L1 cache miss occurs, the request is routed through the CCU to the shared L2 cache. In the event of an L2 cache miss, the request is forwarded to main memory. The arbiter ensures orderly access to shared resources when multiple cores issue simultaneous requests, maintaining system efficiency and coherence.

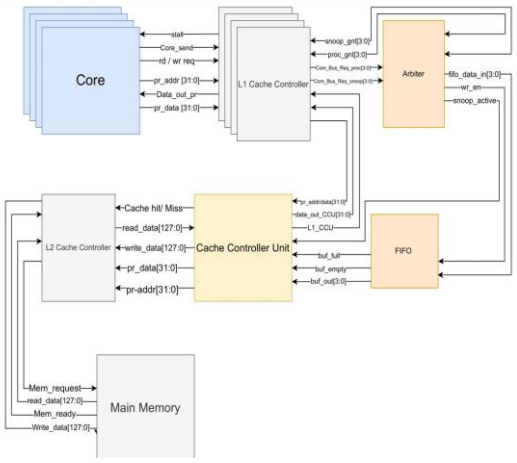

Figure 1: System architecture implementing MESI-based multicore cache coherence.

Figure 1 illustrates the complete architecture of the multi- core cache coherence system based on the MESI protocol. It visually maps the flow of data and control signals between the cores, L1 cache controllers, FIFO buffer, arbiter, Cache Controller Unit (CCU), L2 cache controller, and main memory. Each core operates with its own private L1 cache and communicates with shared components via a snooping bus arbitrated for fairness and exclusivity. The diagram reinforces the previously described functional flow—starting from the core's memory request, traversing through various layers of cache logic, and culminating in main memory access if needed. This architectural layout ensures coherent data sharing, reduced latency, and optimized memory hierarchy coordination across all cores.

3. Results and Verification

This section outlines the verification methodology employed for the multicore cache coherence architecture and presents the corresponding simulation outcomes that validate both functional correctness and MESI protocol compliance.

UVM-Based Verification Environment

To rigorously validate the behaviour of the proposed quad- core system implementing the MESI cache coherence protocol, a comprehensive Universal Verification Methodology (UVM) environment was developed using System Verilog. This environment encapsulates modular agents for each processor core and associated memory components. Each agent comprises a driver, sequencer, and monitor, enabling the generation and collection of transactions in a controlled and reusable manner. A centralized virtual sequencer coordinates test execution across all cores, while a scoreboard—integrating a functional reference model and checker—validates observed results against expected behaviour. The verification testbench also includes protocol monitors for the communication bus and memory subsystem to detect inconsistencies in arbitration or coherence state transitions.

Figure 2 illustrates the structural composition of the UVM testbench. The setup emphasizes scalability and configurability, making it adaptable to varying numbers of cores and coherence protocol parameters. This environment supports both directed and randomized test scenarios, enabling the detection of corner cases and subtle protocol violations.

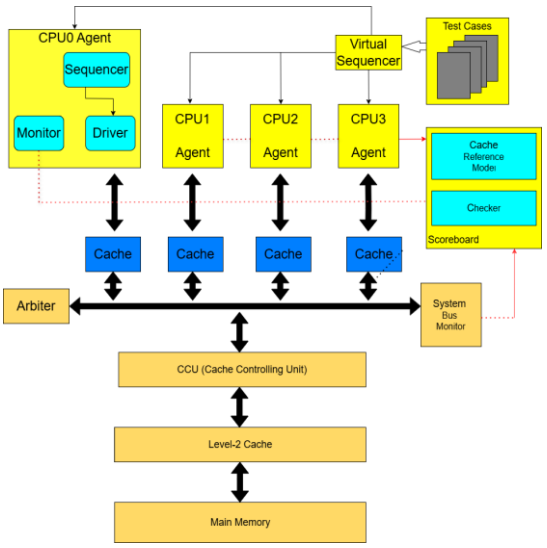

Figure 2: UVM-based verification environment for multicore MESI system.

The primary verification goals included ensuring correct MESI state transitions under concurrent memory operations, validating cache-to-cache transfers initiated by the Cache Coherence Unit (CCU), confirming arbiter behaviour under contention, and establishing overall memory consistency. Simulation results were captured and analysed in waveform viewers to monitor internal signal activity and coherence state propagation across cores.

One of the directed test scenarios involved a simultaneous read request from all four cores to a common memory address. In this case, the coherence logic correctly allowed only the first core to fetch the data from main memory, entering the Exclusive state. Subsequent cores obtained the data via cache-to-cache transfer and transitioned into the Shared state, with no unnecessary memory traffic observed. This not only demonstrated compliance with the MESI protocol but also confirmed optimization of memory bandwidth by avoiding redundant memory accesses. Importantly, no cores entered the Modified or Invalid state, as the transaction involved only read operations.

Across all test scenarios, including randomized stress tests, the system consistently adhered to expected MESI behaviour.

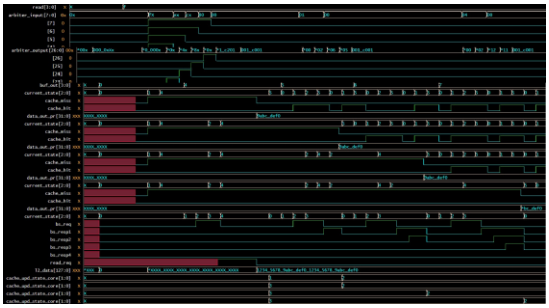

Figure 3: Waveform of all cores reading from the same address.

Arbitration was handled deterministically, ensuring fairness and correctness. Functional coverage analysis reported over 95% coverage of the defined stimulus space, and no deadlocks or assertion failures were encountered during simulation.

These results substantiate the correctness, robustness, and efficiency of the implemented design and highlight the effectiveness of UVM as a structured verification methodology for complex multicore systems.

4. References

[1] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 6th ed. Morgan Kaufmann, 2017.

[2] M. Papadimitriou, “Challenges in Cache Coherence for Multi-Core Processors,” IEEE Transactions on Computers, vol. 65, no. 3, pp. 789–802, 2016.

[3] S. Gochman, A. Mendelson, A. Naveh, and E. Rotem, “The Intel Pentium M Processor: Microarchitecture and Performance,” Intel Technology Journal, vol. 7, no. 2, 2003.

[4] A. Sodani, “Split-Transaction Buses: Design and Challenges,” IEEE Micro, vol. 35, no. 4, pp. 56–65, 2015.

[5] A. Adir, H. Attiya, and G. Shurek, “Verification of Cache Coherence Protocols: A Case Study,” IEEE Design & Test of Computers, vol. 30, no. 2, pp. 42–51, 2013.

[6] D. J. Sorin, M. D. Hill, and D. A. Wood, A Primer on Memory Consistency and Cache Coherence. Morgan & Claypool Publishers, 2011.