Securing and Optimizing Data Flow by Validating Die Level Interconnect Stress with AES XTS 256-bit Encryption and Ring Interconnect Analysis on edge client SoCs

Gowri Shankar Somanjeri
Pankaj Sharma, Ashutosh Mishra, Naveen Urs T R, Aravind Krishnan, Srishti Singh Chauhan

Introduction

Seamless Integration of MKTME into Automated Test Frameworks

Problem Statement
- Testing tools exist but lack full encrypton validation
- MKTME integration is not streamlined

Current Limitations
- Only basic MKTME usage supported
- No smooth handling of key aliases

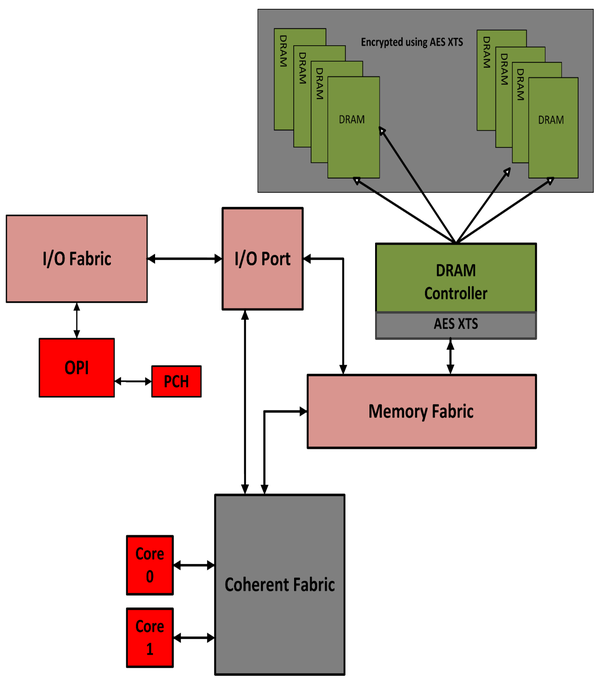
Proposed Solution
- Wrapper tool converts front-end test lines → MKTME-compatible lines
- Enables advanced validation across system configurations

Impact
- Comprehensive validation of encryption features
- Robust, automated testing pipeline
- Future-proof for evolving test needs

Proposed Methodology/Advantages

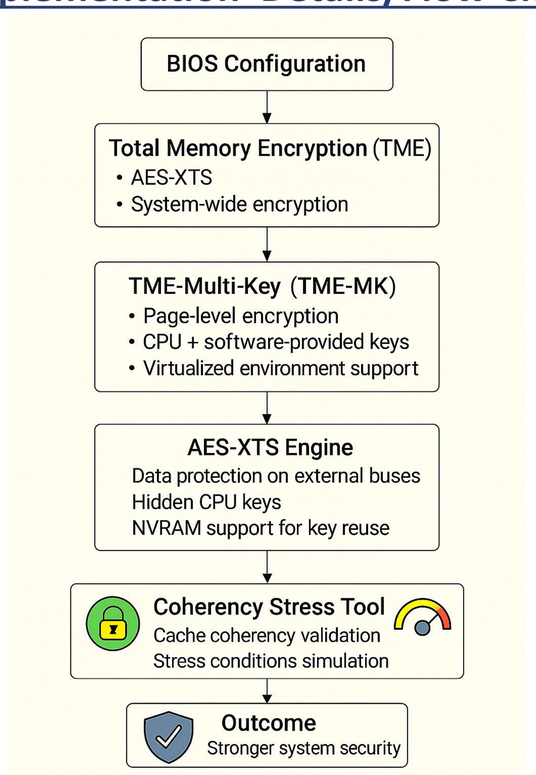
- Existing test frameworks need memory encryption integration for full validation; Front-end Tool lacks direct MKTME support.
- MKTME incorporation is essential to verify encryption features.
- A wrapper adapts test lines for MKTME, enabling broader and advanced testing.
- The tool assigns unique KeyIDs and MKTME commands to memory targets; threads inherit these encryption attributes.
- Key aliasing lets multiple MKTME keys map to the same memory location, testing cache coherency and eviction.
- Streamlines MKTME integration, improves thoroughness, and supports realistic scenarios like key aliasing and inheritance.

Coherent Fabric Overview

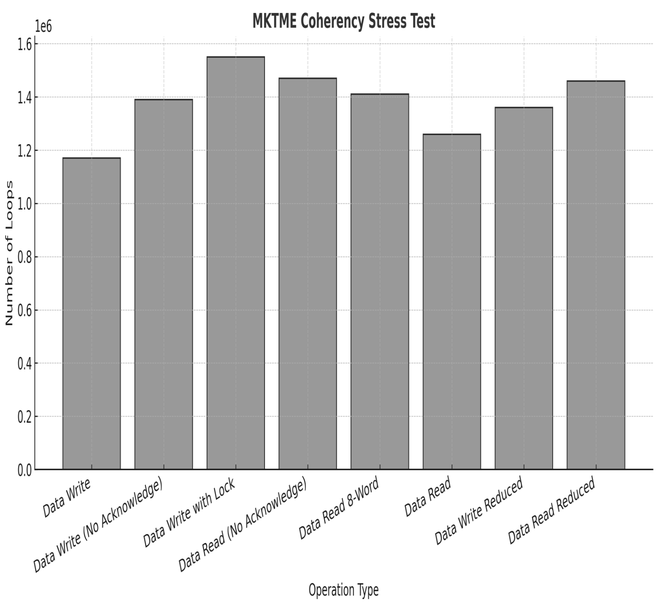


- The Coherent Fabric is a scalable interconnect and caching solution in SoC architecture that efficiently links cores, cache slices, and system agents within a package.
- Using a ring-based topology, it manages data movement, maintains cache coherence, and handles memory requests.
- Its modular design supports advanced power management and error reporting, providing flexibility, scalability, and robust performance for modern SoC platforms.

Implementation Details/Flow Chart



Results Table



- **Seamless Integration:** A wrapper solution around the Front-end Tool converts test lines into MKTME-compatible lines, enhancing the testing process and ensuring comprehensive validation of encryption features.
- **Automated Key Management:** The System coherency Stress tool automates the assignment of MKTME commands, globally applying encryption algorithms and allocating KeyIDs, reducing manual intervention and potential errors.
- **Advanced Testing Capabilities:** Enables key aliasing for complex testing scenarios.

Conclusion

This paper introduces a wrapper-based solution that integrates Multi-Key Total Memory Encryption (MKTME) into system testing frameworks, enabling seamless validation of encryption features across a wide range of configurations. By automating command assignment, applying global algorithms, and supporting key aliasing, the framework simplifies key management while reducing manual effort and potential errors. These enhancements provide greater flexibility, scalability, and accuracy in testing, representing a significant step forward in memory encryption validation and offering robust tools to support secure system deployment in increasingly complex digital environments.

REFERENCES

[1] Intel Corporation, “Intel® Architecture Memory Encryption Technologies Specification,” Rev. 1.0, Sept. 2019. [Online]. Available: https://www.intel.com
[2] S. Gueron and R. Johnson, “AES-XTS: Counter-Mode Encryption for Secure Memory,” IACR Cryptology ePrint Archive, vol. 2010, pp. 1–24, 2010.
[3] Intel Corporation, Intel® 64 and IA-32 Architectures Software Developer’s Manual: Vols. 1–3, Santa Clara, CA, USA: Intel, 2022.

© Accellera Systems Initiative