MEXT

WHITEPAPER | Q1'26

Driving Superior Performance-Per-Dollar with 50% Less DRAM

An analysis of MEXT AI-Powered Predictive Memory™ software performance for signoff

---

Executive Summary

This whitepaper highlights how MEXT Predictive Memory™ software transforms the economics of running EDA workloads like signoff. Signoff workloads are notoriously memory-bound, requiring costly and often over-provisioned DRAM to handle complex data. With MEXT, teams can unlock near-identical performance as all-DRAM baselines using systems with 50% or less DRAM—yielding vastly superior performance-per-dollar. MEXT installation takes < 5 min and requires no infrastructure changes.

A major semiconductor company's testing illustrated the following results. Against a 512GB all-DRAM baseline scenario: equivalent performance as baseline after reducing DRAM by 50% and adding MEXT, near-equivalent performance after reducing DRAM by 62.5% and adding MEXT, and 80% performance after reducing DRAM by 75% and adding MEXT. Against a 1.5TB all-DRAM baseline scenario: 88% performance after reducing DRAM by 50% and adding MEXT.

These results translate to:

- **2.1X** higher perf-per-dollar (512GB baseline, 50% DRAM reduction scenario)
- **2.7X** higher perf-per-dollar (512GB baseline, 62.5% DRAM reduction scenario)
- **3.2X** higher perf-per-dollar (512GB baseline, 75% DRAM reduction scenario)
- **1.76X** higher perf-per-dollar (1.5TB baseline, 50% DRAM reduction scenario)

Introduction

Signoff is one of the most computationally demanding steps in EDA workflows, requiring precise modeling of timing paths across billions of transistors and interconnects. These analyses involve massive, highly interconnected data structures that strain system memory, often pushing DRAM capacity and bandwidth to their limits. As design complexity grows with each process node, the sheer volume of timing data leads

to frequent paging, reduced throughput, and underutilized compute resources—creating a significant bottleneck in design turnaround time and overall engineering efficiency. Optimizing memory usage without compromising accuracy has therefore become a critical challenge, directly impacting schedule predictability, tool utilization, and total compute cost.

Hardware Configuration

- 1x AMD EPYC 9755 128-core processor
- 12x 128GB 6300MT/s RAM
- KIOXIA NVMe drive

Software Stack

- Red Hat Enterprise Linux 8.10
- Linux Kernel 6.17.1
- MEXT Predictive Memory™
- Signoff tool

.

MEXT AI-Powered Predictive Memory™

DRAM Challenges

Server memory (DRAM) comes with some major challenges. First, it is now the most expensive data center component, with cost-per-bit already up 3.5X in the past 2 quarters and showing no signs of slowing. Second, predicting how much is needed for a particular job is very difficult, and getting it wrong can result in out-of-memory (OOM) errors and job failures; as a result, most teams over-provision to avoid the “memory cliff”. Third, for larger memory systems, the granularity of available memory sizes is quite coarse; even if the exact requirement is known, it’s impossible to fine-tune the purchased memory to the requirement plus a small buffer (e.g., 10%). Lastly, even if the precisely-correct memory size is somehow implemented, utilization regularly drops to 50% or lower (as demonstrated by Meta and others). For certain EDA applications specifically, semiconductor companies have witnessed that only 15-30% of memory pages are “hot” (actively utilized) at any given time period. Even if these memory pages are not being actively used, they are still required to be present for the system to operate. In short, DRAM is extremely expensive, often over-provisioned, hard to fine-tune, and poorly-utilized.

How MEXT Works

MEXT's software-only solution makes flash storage appear as DRAM-like memory to the OS.

Here's how it works:

- MEXT continuously monitors which memory pages in DRAM actively being utilized, or hot, and which have gone cold
- MEXT offloads the cold memory from DRAM to flash
- MEXT leverages AI to mitigate the effects of flash latency and keep the system performant (via the MEXT Predictive Memory™ Engine)
    - This engine continually predicts which offloaded pages might be requested by the application soon (in other words, which pages are likely to soon go from cold to hot), and transparently moves them back into DRAM before the requests are even made. The result? The application stays performant because from its perspective, the relevant memory pages are always already resident in DRAM.

Value

In this way, MEXT addresses the major DRAM challenges outlined above. First, by leveraging low-cost flash as memory (50X cheaper than DRAM), MEXT enables much more cost-effective computing. Second, MEXT prevents OOM crashes by enabling applications to go out to flash-as-memory if DRAM ever runs out. Third, MEXT enables more precise right-sizing of memory sizes to true requirements. Lastly, MEXT solves the utilization issue by intelligently tiering across DRAM and flash to keep DRAM mostly all hot.

To quantify this: our testing with various customers and partners illustrates that MEXT enables up to 50% lower computing costs (by delivering equivalent application performance on lower-DRAM systems) or 2-4X more memory capacity on existing hardware / within the existing budget.

Seamless Implementation

MEXT is a patent-pending, software-only solution that works with any configuration: on-premises or in any cloud, with any processor, across virtualized / bare-metal / containerized environments, and requiring no changes to the OS or applications. Installation takes less than 5 minutes.

MEXT Solution Components

The MEXT Predictive Memory™ solution consists of 3 primary components: the MEXT Driver, the MEXT Predictive Memory™ Engine, and the MEXT View™ Observability Platform.

MEXT Driver

The MEXT Driver is a dynamically loadable kernel module (which does not alter the standard Linux kernel) that sends process and memory page telemetry data to the MEXT Predictive Memory™ Engine.

MEXT Predictive Memory™ Engine

The MEXT Predictive Memory™ Engine is a user-space process that feeds predictions of which memory pages should be pushed from flash to DRAM—making predictions / inferences in under a fraction of a second. It runs entirely on the local Linux operating system (on a single CPU core) and does not require a GPU.

It was inspired by modern AI techniques based on neural networks. Instead of using these techniques to predict words or natural language patterns (like ChatGPT does), it predicts sequences of future memory page accesses. It consists of a family of models that work together, combining extremely lightweight heuristic predictors with more powerful neural-network models. For any given workload, it automatically adjusts to use the model or group of models that performs best. Continuous observation of which predicted pages were actually used by the application also enables the engine to acquire real-time feedback regarding model accuracy, supporting ongoing adaptation and self-optimization.

MEXT View™ Observability Platform

MEXT also provides a user-space application called MEXT View™ that provides observability / visualization tools to help customers profile their workloads—illustrating how much memory their applications are using at any given time and what portion of this memory is hot / warm / cold. All cold memory pages are good candidates for optimization by MEXT Predictive Memory™ software. MEXT View™ also provides insight into the ongoing prediction accuracy of the MEXT Predictive Memory™ Engine.

.

Methodology

The testing was conducted by running the baseline test using a 100% DRAM system with no MEXT Predictive Memory™ enabled and capturing the elapsed time for the test. Then, the system memory was altered via a Linux GRUB parameter (“Mem”) to cut the memory presented to the operating system (either by 50%, 62.5%, or 75%, depending on the test). Subsequently, an amount of MEXT Memory™ (MEXT software + flash) equal to the DRAM reduction was enabled in order to return the total amount of system

memory to the amount originally present for the baseline test. The test was once again executed in this MEXT-enabled configuration and the result elapsed time was noted.

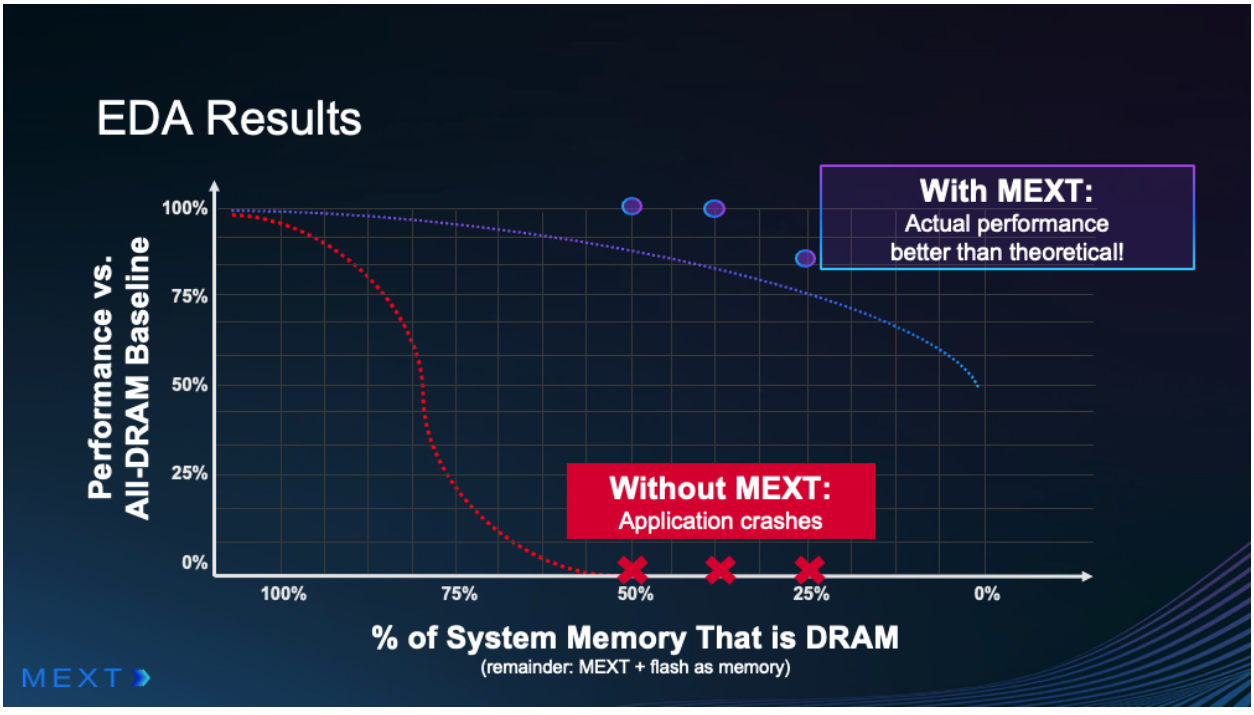
Results

512GB Scenario: Performance

The following results were observed for the 512GB baseline configuration:

Test Configuration	Time (Seconds)
512GB Baseline	14,864s
50% DRAM Reduction **without MEXT**	Out-of-memory crash
50% DRAM Reduction (256GB RAM, 256GB MEXT Memory™)	13,951s
62.5% DRAM Reduction **without MEXT**	Out-of-memory crash
62.5% DRAM Reduction (192GB RAM, 320GB MEXT Memory™)	14,813s
75% DRAM Reduction **without MEXT**	Out-of-memory crash
75% DRAM Reduction (128GB RAM, 384GB MEXT Memory™)	18,592

The normalized performance results are illustrated here:



512GB Scenario: Performance-Per-Dollar

If we assume the cost of systems scales reasonably according to the amount of DRAM within them (i.e. 50% DRAM reduction = 50% cost reduction, etc.), these performance results translate to the following performance-per-dollar improvements:

- **2.1X** higher perf-per-dollar (50% DRAM reduction scenario)
- **2.7X** higher perf-per-dollar (62.5% DRAM reduction scenario)
- **3.2X** higher perf-per-dollar (75% DRAM reduction scenario)

1.5TB Scenario: Performance

The following results were observed for the 1.5TB baseline configuration:

Test Configuration	Time (Seconds)
1.5TB Baseline	98,582s
50% DRAM Reduction **without MEXT**	Out-of-memory crash
50% DRAM Reduction (768GB RAM, 768GB MEXT Memory™)	112,069

1.5TB Scenario: Performance-Per-Dollar

If we assume the cost of systems scales reasonably according to the amount of DRAM within them (i.e. 50% DRAM reduction = 50% cost reduction, etc.), these performance results translate to the following performance-per-dollar improvements:

- **1.76X** higher perf-per-dollar (50% DRAM reduction scenario)

Conclusion

MEXT Predictive Memory™ offers semiconductor companies a powerful path to improve compute efficiency for signoff: this means more runs within the same budget, yielding better-designed chips. The results also mean that if MEXT is added to existing system configurations (with no DRAM reductions), the available memory capacity can be dramatically extended without any new hardware investment. By intelligently turning flash into an extension of DRAM, MEXT effectively multiplies available system memory, enabling much larger datasets to be analyzed in memory rather than spilled to disk. This expanded capacity directly translates to higher throughput, improved scalability, and faster design closure. With MEXT, organizations can push beyond the memory limits of their current infrastructure, unlocking additional performance from every server and accelerating time-to-results.