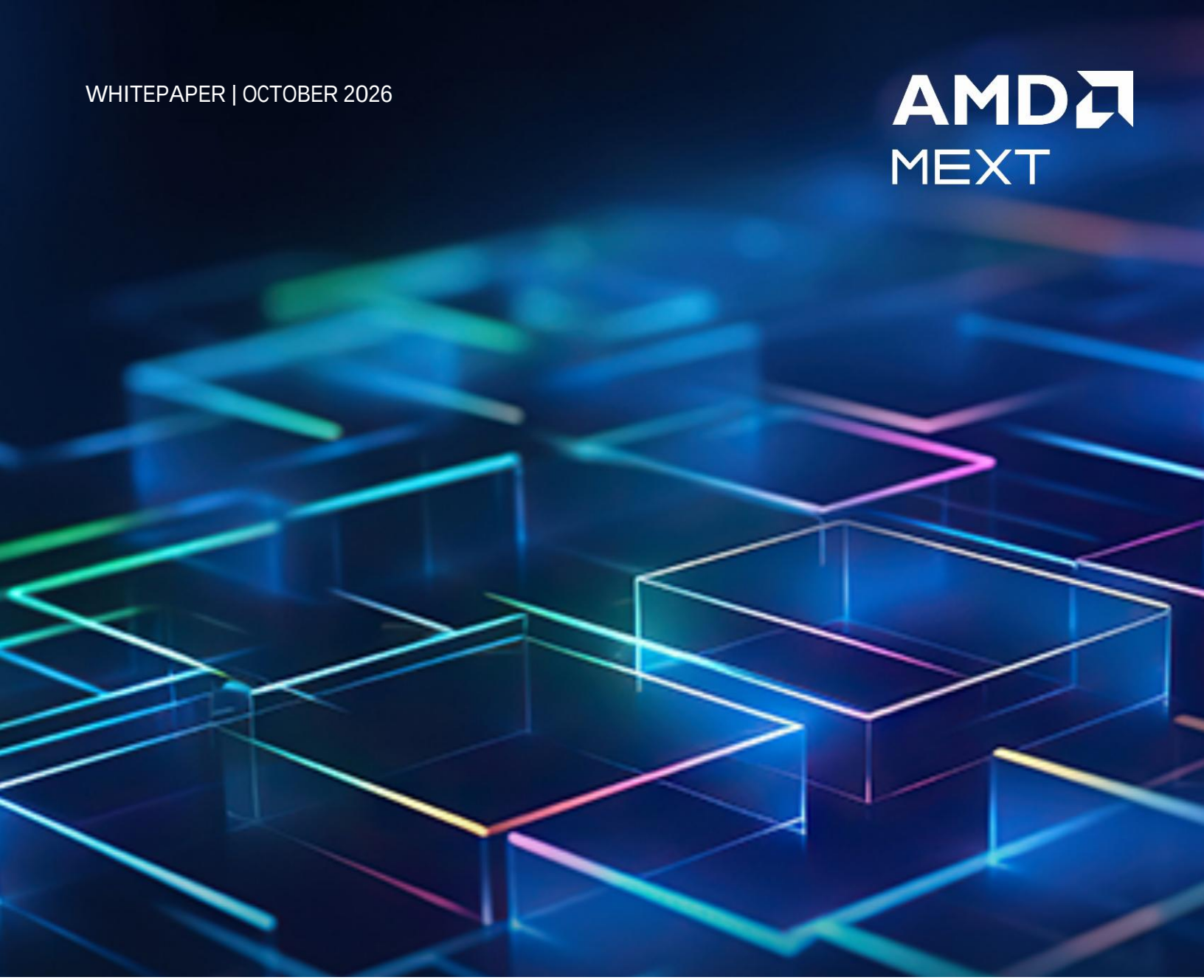
WHITEPAPER | OCTOBER 2026



DRIVING SUPERIOR PERFORMANCE PER-DOLLAR WITH 50% LESS DRAM FOR STATIC TIMING ANALYSIS

An analysis of AMD MEXT software performance for Synopsys® PrimeTime®

DISCLAIMER

©2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, AMD EPYC, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Synopsys and PrimeTime are trademarks or registered trademarks of Synopsys, Inc. PCIe is a registered trademark of PCI-SIG Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies. Supported features may vary by operating system. Please confirm with the system manufacturer for specific features. No technology or product can be completely secure.

EXECUTIVE SUMMARY

This whitepaper highlights how AMD 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 comparable performance as all-DRAM baselines using systems with half the physical DRAM – yielding vastly superior performance-per-DRAM-dollar.

Testing conducted by the AMD EDA team illustrated the following results: Against a 1.5TB all-DRAM baseline scenario, the MEXT-enabled configuration with 50% less physical DRAM (768GB) yielded approximately 87% of the performance vs baseline, translating to 1.74x higher performance-per-DRAM-dollar.[1]

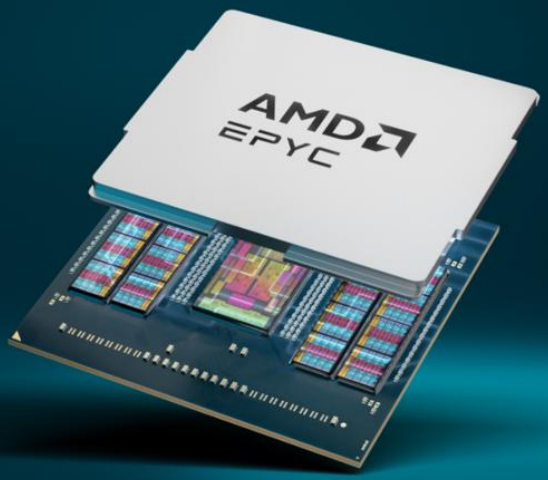
50%	~87%	1.74x
less physical DRAM	of all-DRAM performance	higher performance-per-DRAM-dollar

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

Processor	1x AMD EPYC 9755 128-core processor
Memory	12x 128GB 6400MT/s RAM
Storage	KIOXIA NVMe drive



SOFTWARE STACK

Application	Synopsys PrimeTime
Memory Software	AMD MEXT Predictive Memory software
Kernel	Linux Kernel 6.17.1
Operating System	Red Hat Enterprise Linux 8.10

AMD MEXT PREDICTIVE MEMORY SOFTWARE

How It Works

The MEXT 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 are actively being utilized (hot) and which have gone cold
- MEXT offloads the cold memory from DRAM to flash
- MEXT AI engine mitigates the effects of flash latency to keep the system performant
  - AI 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)
  - AI engine transparently moves those pages back into DRAM before the requests are even made
  - As a result, the application stays performant because from its perspective, the relevant memory pages are always already resident in DRAM

THE MARKET: DRAM CHALLENGES

Server memory (DRAM) comes with some major challenges.

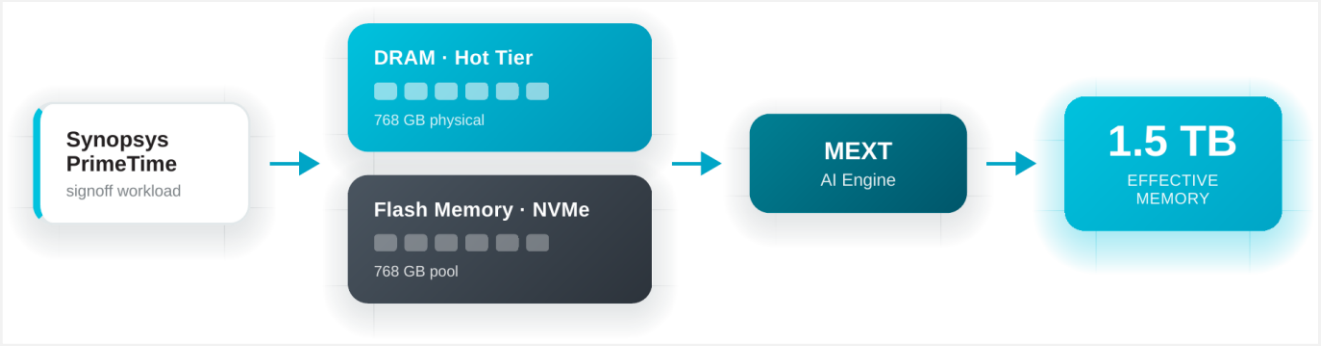
First, it is now the most expensive data center component, with cost-per-bit increasing dramatically and showing no signs of slowing.[2]

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).[3]

In short, DRAM is extremely expensive, often over-provisioned, hard to fine-tune, and usually poorly-utilized.



MEXT Value

In this way, MEXT addresses the major DRAM challenges outlined above.

First, by leveraging low-cost flash as memory (50x cheaper than DRAM)[4], 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.

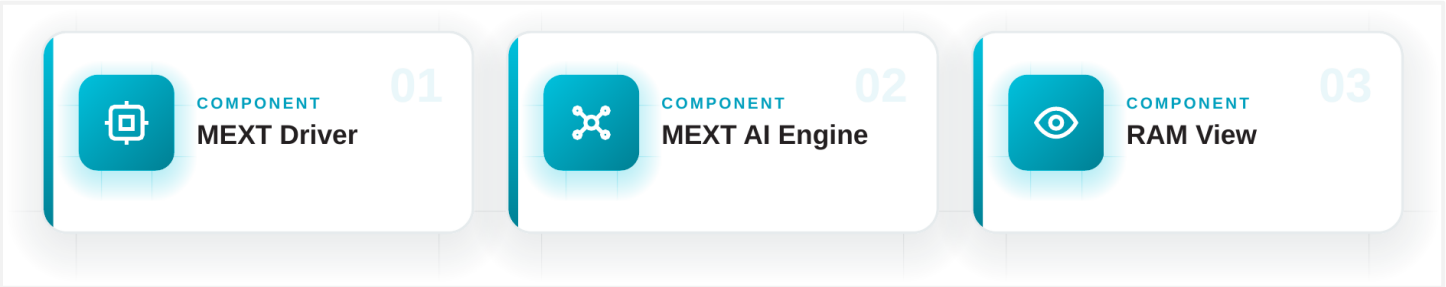
To quantify this: our testing with various customers and partners illustrates that MEXT enables up to 50% lower computing costs (by delivering equivalent application performance on lower-DRAM systems) or up to 4x memory capacity on existing hardware, within the existing budget (through MEXT + flash operating as memory).

Seamless Implementation

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

AMD MEXT SOLUTION COMPONENTS

The solution consists of 3 primary components: the MEXT Driver, the MEXT AI Engine, and the RAM View Observability Platform.



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 AI engine.

MEXT AI Engine

The AI 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.

RAM View Observability Platform

RAM View is an observability / visualization tool that illustrates how much memory the application is consuming including real-time insights into page hotness / warmness / coldness. It illustrates the proportion of memory pages that were accessed <15s ago, <1min ago, <2min ago, <3min ago, <5min ago, and >5min ago.

METHODOLOGY

The testing was conducted by running the baseline test on a 1.5TB DRAM system with no MEXT software enabled and capturing the elapsed time for the test. Then, the system memory was altered via a Linux GRUB parameter ("Mem") to cut the DRAM presented to the operating system by 50% (representing a 768GB DRAM system). Subsequently, 768GB of MEXT memory (MEXT software + flash) was given back to the system to bring the total (effective) system memory back up to 1.5TB. The test was once again executed in this MEXT-enabled scenario and the result elapsed time was noted.

RESULTS

Test Configuration	Time (Seconds)
1.5TB Baseline	~98,000s (3-run average)
50% DRAM Reduction without MEXT	Out-of-memory without a memory safety net
50% DRAM Reduction (768GB RAM, 768GB MEXT Memory)	~112,000s (3-run average)

Performance-Per-DRAM Dollar

The MEXT-enabled configuration resulted in an average slowdown of 13% across the 3 runs. In other words, the MEXT-enabled configuration achieved about 87% of the raw performance vs. the all-DRAM baseline.

If we take the normalized performance at 50% the DRAM cost, this translates to a **performance-per-DRAM-dollar boost of 1.74x** with MEXT.[1]

CONCLUSION

MEXT 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 significantly extended without any new hardware investment. 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.

ENDNOTES

1. 9xx5-289a: AMD Internal testing as of 10/1/2025 on Synopsys PrimeTime, an EDA signoff workload, on an AMD EPYC 9755 CPU with 1.5TB DRAM only memory as the baseline vs. the same AMD EPYC 9755 CPU using 768GB DRAM and 768GB MEXT memory pooled from a 3.84TB Samsung NVMe drive.

   Testing measured performance, measured in seconds (lower is better) in matched NPS1/NPS2 pairs.

   System configurations:
   AMD EPYC 9755
   - AMD Volcano CRB reference board. BIOS RVOT1004E.
   - 1P, 128 cores per socket, 24 memory channels
   - NVMe for MEXT memory: Samsung MZQL23T8HCLS-00A07, 3.84 TB, PCIe Gen4x4, TLC
   - Base/Boost frequencies of 2.7 GHz/4.1 GHz, respectively
   - L1/L2/L3 cache of 12 MiB/ 256 MiB/ 1 GiB, respectively
   - NPS1/NPS2
   - 1.5TB Memory using 12x 128 GB DDR5 DIMMs with 6400 MT/s speeds
   - OS version: Red Hat Enterprise Linux 8.10
   - Kernel: Linux Kernel 6.17.1
   - GCC version: 8.5.0 (RHEL 8.10 system default)
   - SMT: off
   - All other settings default

   Performance per DRAM dollar reflects normalized performance divided by the relative DRAM capacity used and does not account for NVMe, MEXT licensing, other system costs, or total cost of ownership.

2. CounterPoint Research: https://counterpointresearch.com/en/reports/memory-price-tracker-feb-2026
3. Meta paper: https://dl.acm.org/doi/10.1145/3746651
4. Citrini Research: https://www.moomoo.com/news/post/71585198/citrini-amd-and-apple-both-bet-on-flash-memory-as