ECHO's legacy: smarter multicore processors for a faster and greener future

Alberto Ros
University of Murcia, Spain

Adobe Stock, generated with AI © *Sahriar*

The **ECHO project** (Extending Coherence for Hardware-Driven Optimisations in Multicore Architectures), funded by an ERC Consolidator Grant, concluded in January 2026. ECHO reimagines multicore processors to deliver faster, greener, and easier-to-program computers. With highly accurate mechanisms and a vision of high-performance, non-speculative execution, ECHO minimises wrong speculation, where processors guess ahead but often waste energy on wrong hunches. This innovative approach offers major gains in performance and efficiency for everyday devices like smartphones and powerful servers executing a wide variety of workloads, from scientific simulations to artificial intelligence (AI) applications.

Previous releases of this journal brought the general public updates on ECHO's initial phase (Ros, 2020) and mid-term progress (Ros, 2023). Now it is time to showcase the project's final contributions, focusing on tackling speculation inefficiencies across 4 key areas: hardware data prefetching; memory dependence prediction and disambiguation; non-speculative out-of-order writes; and non-speculative hardware transactional memory. These advances pave the way for processors that speculate less but perform better—boosting speed while saving energy.

Accurate hardware data prefetching

In processors, programs constantly read data from slow main memory into fast on-chip caches. Prefetching predicts the data the processor will need and when, and fetches them early enough into the cache. When done right, it hides memory delays and speeds up programs dramatically, but poor guesses lead to cache pollution and wasted energy.

In fact, traditional methods favour performance and often overfetch. ECHO's Berti prefetcher (Navarro-Torres et al., 2022), its flagship contribution in this area, excels by tracking local deltas (distances in memory addresses between accesses by the same instruction). This ensures accurate learning even when accesses are seen out of order due to the processor's execution engine. Berti achieves superior accuracy, thus saving a considerable amount of expensive data movement.

To transfer this technology to industry, ECHO secured an ERC Proof of Concept Grant (Berti-Chip). The team simplified Berti's mechanisms while maintaining high accuracy and performance (Navarro-Torres et al., 2025). Berti proved itself the state-of-the-art prefetcher, ushering in a new era: its impact shone at the 4th Data Prefetching Championship (DPC-4), where all 8 contributions used Berti (or an extension) at the first-level cache.

ECHO's evolved BertiGo (Singh, 2026) clinched first place at DPC-4 by filtering useless prefetches and leveraging context information to discover more accurate deltas. BertiGo dominated all 3 contest metrics (full bandwidth, limited bandwidth, and multicore), delivering on full bandwidth 17.2% average speedup over the organisers' best state-of-the-art prefetcher—and rising to 27.7% on machine learning benchmarks, demonstrating its power on emerging AI applications.

Accurate memory dependence prediction and disambiguation

When a program runs, it constantly reads and writes data in memory. Sometimes a read (load) must wait for a previous write (store) because it needs the value that is being written; other times, the read is completely independent and could have been done earlier. The problem is that the processor does not know this for sure until it has computed all the memory addresses, which comes late in the pipeline. Memory dependence prediction guesses if one load depends on a prior store, such that dependent loads wait for the store to execute. Prediction errors cause the speculative work to be discarded.

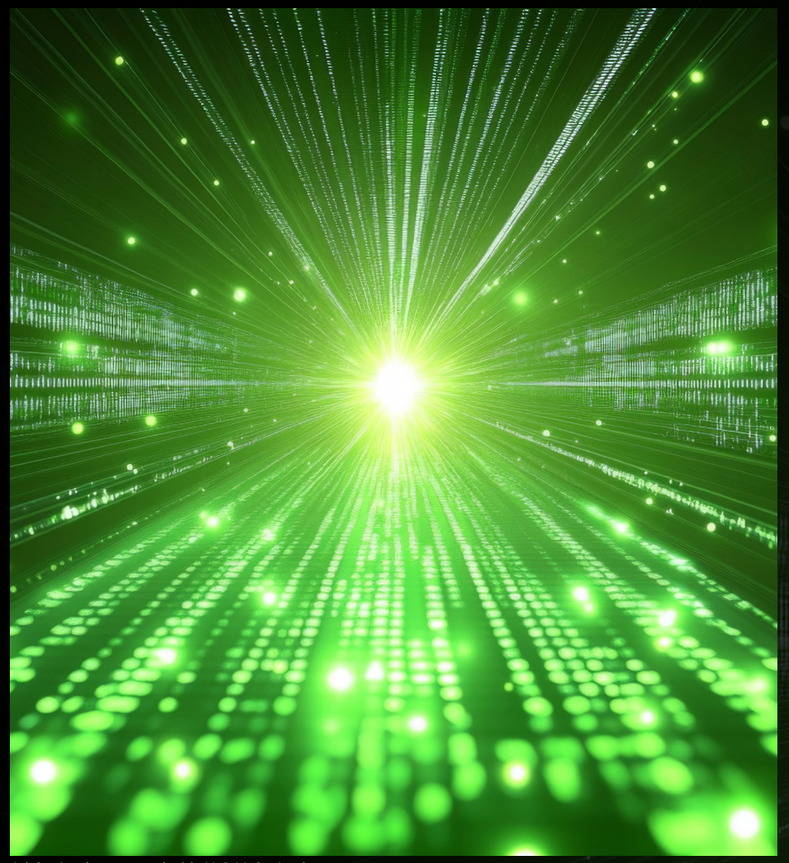
PHAST (Kim and Ros, 2024) proposes a new way to predict memory dependencies by learning them with exactly the right context for each load, rather than using arbitrary history lengths.

Conventional memory dependence predictors often track long histories of branches or memory events, which increases storage, pollutes predictor tables, and still leaves many mispredictions. PHAST's key insight is that each load really only needs to 'remember' its path to its youngest conflicting store. Using this idea, PHAST dynamically finds, per conflicting load-store pair, the minimum history length that uniquely identifies its path and then uses it as the training context. This path-aware approach gives PHAST near-ideal prediction accuracy with much lower storage than prior state-of-the-art schemes. In evaluations, a practical PHAST implementation with a modest hardware budget comes within a very small margin of an ideal predictor. The work was recognised with a Best Paper Honourable Mention, underscoring its impact.

A better alternative than stalling the load on dependencies is memory bypass, where the processor grabs data directly from a recent store instead of waiting for that store to execute, thus speeding execution. In this context, we developed MASCOT (Mose et al., 2025) using context-sensitive tables that track store-to-load distances. Its prediction breakthrough is learning both dependencies and non-dependencies and tracking both usefulness confidence and bypass opportunities. This dual learning delivers ultra-low mispredictions, enabling aggressive bypassing without costly squashes.

Non-speculative out-of-order writes

Intel and AMD processors follow x86's total store order (TSO) rule: writes to memory must become visible to other cores in exact program order. Modern processors speed things up by executing stores out-of-order internally, but TSO's store buffer ensures they drain to the shared cache sequentially. If another thread's load sees these writes out of order, it triggers costly squashes and re-executions, discarding useful work again.



Adobe Stock, generated with AI © *Vader Stocker*

A central goal of ECHO is to answer the following question: Can processors execute writes out-of-order while: 1) still obeying x86's strict rules, i.e. other threads always see writes in program order; and 2) writes execute non-speculatively (no rollbacks)?

ECHO's early ROOW approach (Singh, Jimborean, and Ros, 2020) used compiler annotations to mark 'safe regions' for out-of-order writes. The latest breakthrough, temporary unauthorised stores (TUS) (Cebrian, Jahre and Ros, 2024), eliminates compiler help entirely and accomplishes the mentioned goal. TUS writes stores to cache immediately (out-of-order!) but uses smart hardware checks to ensure other threads see them in the correct program sequence. A predefined global order of writes helps to decide on store priorities to avoid program deadlocks. This delivers non-speculative out-of-order writes all the time, with large speedups in multi-threaded applications.

Non-speculative hardware transactional memory

Hardware transactional memory (HTM) is like a 'try-before-you-buy' system for parallel programming. Instead of using locks that block other threads, HTM is a speculative solution that lets multiple threads work concurrently on shared data. If no conflicts occur (another thread did not access the same data), the transaction commits atomically: all changes become visible at once, as if they happened instantly. But if a conflict is detected, the hardware aborts the transaction, rolls it back, discards speculative changes, and retries (often multiple times). Examples of HTM are Intel's TSX and ARM's TME.

The other key goal of ECHO is to achieve non-speculative HTM, that is, executing transactions concurrently but never requiring the whole transaction to be aborted.

Our recent proposal, CLEAR (Gomez-Hernandez, 2024), achieves that goal by bounding the execution of some transactions to a single retry, eliminating the endless retries that plague traditional HTM. In other words, the second time the transaction executes, it does so non-speculatively and with guarantees of completion. CLEAR uses the first speculative execution of a transaction to gather the exact memory footprint and information about the mutability of that transaction. For immutable transactions, i.e. those that guarantee that the memory footprint does not change across executions, the transaction (if aborted on the first execution) can initiate the retry on the second, but instead of retrying speculatively, CLEAR switches to non-speculative execution, locking the accessed cache lines in a global order (to avoid deadlocks), thus achieving one retry maximum for immutable transactions. Other threads see the locked lines as 'busy', waiting politely. Once complete, changes are committed atomically via the cache coherence protocol. Our results show more than 35% performance improvements over Intel TSX.

Still, mutable transactions can suffer aborts. Our recent CHATS (Nicolás-Conesa et al., 2024) solves this problem by smartly chaining transactions, instead of aborting them. CHATS monitors transaction conflicts and chains the transactions based on their dependencies. A 'parent' transaction absorbs conflicts gracefully, letting 'child' transaction complete successfully. Committed child results then become visible to the parent atomically. This chaining boosts concurrency without locks or endless retries, achieving speedup of more than 40% compared to Intel TSX. CHATS makes transactional memory practical for real parallel applications and is a perfect ECHO complement to CLEAR's single-retry breakthrough.

Conclusion

ECHO delivered on its bold promise: multicore processors that speculate less but perform dramatically better. From BertiGo's championship-winning prefetching to TUS's non-speculative out-of-order writes and CLEAR's single-retry transactions, these innovations slash energy waste while making parallel programming radically simpler.

The ideas developed at the University of Murcia now stand ready for industry adoption. As computing demands explode with AI and big data, ECHO's legacy will shape the next generation of faster, greener processors powering everything from smartphones to supercomputers.

References

Cebrian, J.M., Jahre, M. and Ros, A. (2024) 'Temporarily unauthorized stores: write first, ask for permission later', in *Proceedings of the 57th IEEE/ACM International Symposium on Microarchitecture (MICRO 2024)*. IEEE, pp. 810–822. Available at: https://doi.org/10.1109/MICRO61859.2024.00065.

Gómez-Hernández, E.J., Cebrian, J.M., Kaxiras, S. and Ros, A. (2024) 'Bounding speculative execution of atomic regions to a single retry', in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*. ACM, pp. 17–30. Available at: https://doi.org/10.1145/3622781.3674176.

Kim, S.S. and Ros, A. (2024) 'Effective Context-Sensitive Memory Dependence Prediction', *30th Symposium on High Performance Computer Architecture (HPCA)*, pp. 515–527. March 2024. Available at: https://doi.org/10.1109/HPCA57654.2024.00045.

Mose, K.H. et al. (2025) 'MASCOT: Predicting memory dependencies and opportunities for speculative memory bypassing', in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA 2025)*, 1–5 March 2025, pp. 59–71. Available at: https://doi.org/10.1109/HPCA61900.2025.00016.

Navarro-Torres, A. et al. (2022) 'Berti: An Accurate Local-Delta Data Prefetcher', *55th International Symposium on Microarchitecture (MICRO)*, pp. 975–991. Available at: https://doi.org/10.1109/MICRO56248.2022.00072.

Navarro-Torres, A. et al. (2025) 'A complexity-effective local delta prefetcher', *IEEE Transactions on Computers*, 74(5), pp. 1482–1494. Available at: https://doi.org/10.1109/TC.2025.3533086.

Nicolás-Conesa, V. et al. (2024) 'Chaining transactions for effective concurrency management in hardware transactional memory', in *Proceedings of the 57th IEEE/ACM International Symposium on Microarchitecture (MICRO 2024)*, 2–6 November 2024. IEEE, pp. 840–855. Available at: https://doi.org/10.1109/MICRO61859.2024.00067.

Ros, A. (2020) 'Changing the future in computers', *The Project Repository Journal*, 6, pp. 126–128. Available at: https://www.europeandissemination.eu/project-repository-journal-volume-6-july-2020/11055.

Ros, A. (2023) 'Towards faster, greener and easier to program computers', *The Project Repository Journal*, 16, pp. 34–37. Available at: https://doi.org/10.54050/PRJ1619828.

Singh, S., Navarro-Torres, A. and Ros, A. (2026) 'Pushing the limits of the Berti prefetcher', *4th Data Prefetching Championship (DPC-4)*, February. Available at: https://sites.google.com/view/dpc4-2026/program/main-program.

Singh, S., Jimborean, A. and Ros, A. (2020) 'Regional Out-of-Order Writes in Total Store Order', *PACT '20*: Proceedings of the ACM International Conference on Parallel Architectures and Compilation Technique. Available at: https://doi.org/10.1145/3410463.3414645.

ECHO

PROJECT SUMMARY

The ERC Consolidator Grant project ECHO—Extending Coherence for Hardware-Driven Optimisations in Multicore Architectures—aims to change the events that occur in multiprocessors such that predictions or decisions being made by the processing units will find the best possible outcome in the future. This way, large performance and energy improvements are expected in future computers leveraging ECHO concepts.

PROJECT PARTNERS

The ECHO project is based at the Faculty of Computer Science of the University of Murcia. The Computer Engineering department has a long tradition in computer architecture research and collaborates with several renowned research teams from Europe, America, and Asia.

PROJECT LEAD PROFILE

Alberto Ros is full professor in the Computer Engineering Department at the University of Murcia, Spain.

Funded by the Spanish government to conduct PhD studies, he received PhD in computer science from the University of Murcia in 2009. He held postdoctoral positions at the Universitat Politècnica de València and Uppsala University. He received an European Research Council (ERC) Consolidator Grant in 2018 to improve the performance of multicore architectures, and an ERC Proof of Concept Grant in 2024.

Working on cache coherence, memory hierarchy designs, memory consistency, and processor microarchitecture, he has co-authored more than 100 peer-reviewed articles.

He has been inducted into the ISCA Hall of Fame and the MICRO Hall of Fame.

He is IEEE Senior member.

PROJECT CONTACT

ECHO: Extending Coherence for Hardware-Driven Optimisations in Multicore Architectures
Facultad de Informática, Campus de Espinardo, 30100.

aros@um.es
webs.um.es/aros

FUNDING

This project has received funding from the European Research Council (ERC) under the European Union's research and innovation programme (Grant agreement No. 819134 and 101158023).

Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the granting authority can be held responsible for them.