RETROSPECTIVE: A Performance Comparison of Contemporary DRAM Architectures (ISCA 1999)

Bruce L Jacob, *Fellow, IEEE* and Trevor N Mudge, *Fellow, IEEE*

I. Historical Perspective

BELIEVE it or not, there once was a time, not so very long ago, when the computer's memory system — i.e., everything beyond the last level cache — was considered to be an unknown, a mysterious black box ... and all people could do was to sit about, wringing their collective hands, and complain about the memory system's deleterious effect on their computer's performance.

Caches were, of course, well understood, but everything *beyond* the cache, in particular the main memory system, was known to be problematic but was yet ignored.

Numerous ISCA papers were written with the first two sentences being, and we paraphrase here, something like the following:

> The memory system is the problem; it slows down the microprocessor. Therefore, we propose the following modification/s to the microprocessor ...

In other words, rather than addressing the problem — the problem being the memory system — researchers were focused on making faster, better, more powerful microprocessors, and in those instances where they were concerned with the problem of the memory system, they invented microprocessor-based solutions that *tolerated* the problems caused by slow memory.

At the time, microprocessors were running hundreds to thousands of times faster than memory, thereby restricting system performance severely. This fact was famously highlighted by many, including Wulf & McKee's 1995 "memory wall" paper [1] and Dick Sites' 1996 call to arms ("It's the memory, stupid!" ... an attempt to rally his fellow designers at Digital) [2].

However, despite all the attention paid to this well known and widely acknowledged problem, nobody was focusing their research on the memory system itself. While nearly everyone complained about the memory system, nobody was studying it directly; for instance, there were no papers describing what DRAM memories behaved like or quantifying their effects. For context, in the mid 1990s, the most recent book published on memory systems was Dick Matick's excellent and comprehensive work — copyright 1977, a full twenty years prior [3]. Other books were little more than collections of data sheets.

Such was the state of the union in the mid 1990s, when we began writing this paper: the memory system was a huge, widely acknowledged problem, but few details were known about it. Numerous DRAM architectures were available commercially, but there were no performance studies comparing them. There was a clear need to characterize and understand the problem, and, because there were no accurate models of the DRAM system at that time, we decided to create one ourselves and model the memory system in detail, which would provide us the missing characterization and understanding.

Bruce Jacob is with the Cyber Science Department at the United States Naval Academy, Annapolis, MD, 21402 USA e-mail: bjacob@usna.edu.

Trevor Mudge is with the Electrical Engineering & Computer Science Department at The University of Michigan, Ann Arbor, MI, 48109 e-mail: tnm@umich.edu.

II. Why It Mattered

The following provides some insight into the problem of failing to model memory accurately, why it was important to introduce to the community a better understanding of DRAM and a better DRAM model.

In the 1990s, and in some instances well into the 2000s, simulators did not take into account the behavior of DRAMs or their controllers. Instead, the numerous processor simulators of the day used variations of the following code to emulate the main memory system:

```
if (cache_access(addr) != HIT) {
    cycles += DRAM_ACCESS;
}
```

As one can imagine, this representation of main memory as a constant access time fails to represent the behavior of a real memory system in several significant ways. Note that the model above *can* accurately represent the simplest of systems, in particular single-core/single-thread scalar processors that stall on every cache miss. However, the model becomes increasingly inaccurate as processor designs become more complex, which is exactly what was happening throughout the mid– to late–1990s and beyond.

A few of the ways in which the model above produces an inaccurate simulation result:

- The model implicitly allows an unlimited number of concurrent accesses to memory.
- The model ignores the fact that DRAM access times are dependent on the DRAM state at the time of the access.
- The model fails to account for the memory controller reordering requests, which can occur frequently.

Processor designs became more complex over time, supporting numerous cores, numerous threads per core, increasingly wide-issue pipelines, and non-blocking access to the memory system. These rendered this simple memory-modeling approach increasingly inaccurate. As a result, the simulators of the day (e.g., the late 1990s) began to overestimate performance by significant and increasing amounts — by *orders of magnitude*, in some cases.

The following is just one example. In the late 1990s and early 2000s, the consensus in the computer-architecture community was that implementing data prefetching in server systems was a very good idea. This common wisdom was based on experimental results from research papers that used simplistic models of the memory system. Despite the prevailing wisdom, however, *actual* server systems were routinely shipped with their prefetch hardware *disabled* [4]. This would seem an odd decision, given the academic consensus of the time. However, as Srinivasan shows, using an *accurate* model of the memory system, instead of a simplistic one, reveals that prefetching in server environments fails to improve performance and instead *degrades* it ... which is exactly what the industry knew from experience, if not from simulation.

III. Advances in Memory Systems

That was the scene in the late 1990s. Enter this paper and DRAMsim [5], the first simulator to capture from a systems viewpoint some of the improvements that DRAM manufacturers had started to incorporate in their chips. These improvements included, for the 1999 paper, designs ranging from "evolutionary" to "revolutionary" (labels via S. Przybylski) ... Fast Page Mode, Extended Data Out, Synchronous, Enhanced Synchronous, Synchronous Link, Rambus, and Direct Rambus DRAM architectures.

It should be noted that nearly all of these changes were greatly resisted by the DRAM manufacturers. Their reluctance to innovate arose from the small profit margins of their business.

Our simulations revealed several things: (a) at the time of the paper, "advanced DRAM" technologies were starting to attack the memory bandwidth problem but not the latency problem; (b) the paper noted that bus transmission speed would soon become a primary factor limiting memory system performance; (c) the post-L2 address stream still contains significant locality, though it varies from application to application; and (d) as we moved to wider buses, row access time became more prominent, making it important to investigate techniques to exploit the available locality to decrease access time.

Follow-on work appeared from our own group as well as many others, and, as a critical mass of research on memory systems developed, an appreciation and understanding of the problem spread. In 2000, we performed a study for JEDEC on what was at the time the upcoming DDR2 SDRAM generation [6]. In 2002, we gave a half-day tutorial on DRAM [7]. In 2007, we published a significant reference work on the entire memory system [8]. At the same time, the DRAM industry began to increase memory speed dramatically, ramping up from tens of MHz in the mid 1990s to hundreds of MHz in the 2000 timeframe, corresponding to the introduction of DDR SDRAM, to GHz speeds and more, a decade later with DDR3. Later versions of DRAMsim [9], [10] modeled these advances and others including Hybrid Memory Cube (HMC) and High-Bandwidth Memory (HBM). In 2015, we founded MEMSYS, the *International Symposium on Memory Systems*, which will have its tenth meeting in 2024 (https://memsys.io).

IV. How Did the Future Address the Problems?

Today, we are several generations of DDR beyond, and for our highest-bandwidth needs we have high-bandwidth memory, HBM, which solved the bandwidth problem through packaging: thousands of high-speed data lanes, using silicon instead of a circuit board and FR4 as the substrate.

Beginning with the needs of high-performance computing and then advanced graphics capabilities, the case was made to the DRAM industry that another industry would make use of significant bandwidth (meaning: if you build it, people will buy it). The DRAM industry complied and put out ever-faster parts. AI replaced graphics as the "killer app" demanding significant bandwidth, but the ever-increasing need for data remains.

The problems now facing us are primarily in the domain of modeling. Simulation speeds are such that simulating a significant workload takes a vast amount of time, prompting researchers to use statistical methods of simulating the memory system, instead of cycle-accurate simulation [11]. Interfaces have changed, and these should be studied in detail. As described above, HBM addressed the performance issues facing the systems industry using low tech: packaging. This has been modeled, and it will be interesting to see at what point the modeling and systems integration converge in deciding what is cost-effective for commodity systems. More "revolutionary" interfaces such as CXL must also be modeled and then studied widely, to understand their implications on future systems design.

Quite a bit has happened in the past 25 years, largely because numerous people highlighted the weaknesses of existing systems and pointed out the business sense of fixing them. We can hardly imagine what the next 25 years will bring.

References

[1] W. A. Wulf and S. A. McKee, "Hitting the memory wall: implications of the obvious," *SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995.

[2] R. Sites, "It's the memory, stupid!" *Microprocessor Report*, vol. 10, no. 10, pp. 2–3, Aug. 1996.

[3] R. E. Matick, *Computer Storage Systems & Technology*. New York: A Wiley-Interscience Publication by John Wiley & Sons, 1977.

[4] S. Srinivasan, "Prefetching vs the memory system: Optimizations for multi-core server platforms," Ph.D. dissertation, University of Maryland at College Park, 2007.

[5] D. T. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob, "DRAMsim: a memory system simulator," *SIGARCH Comput. Archit. News*, vol. 33, pp. 100–107, 2005.

[6] B. Davis, T. Mudge, B. Jacob, and V. Cuppu, "DDR2 and low-latency variants," in *Proc. Memory Wall Workshop, held in conjunction with the 27th International Symposium on Computer Architecture (ISCA 2000)*, Vancouver BC, Jun. 2000.

[7] B. Jacob and D. Wang, "DRAM: Architectures, Interfaces, and Systems ... A Tutorial," in *2002 International Symposium on Computer Architecture*, Anchorage AK, May 2002.

[8] B. Jacob, D. T. Wang, and S. W. Ng, *Memory Systems: Cache, DRAM, Disk*. Burlington, MA: Morgan Kaufmann Publishers, Sep. 2007.

[9] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "DRAMsim2: A cycle accurate memory system simulator," *IEEE Computer Architecture Letters*, vol. 10, no. 1, pp. 16–19, 2011.

[10] S. Li, Z. Yang, D. Reddy, A. Srivastava, and B. Jacob, "DRAMsim3: A cycle-accurate, thermal-capable DRAM simulator," *IEEE Computer Architecture Letters*, no. 2, pp. 110–113, 2020.

[11] S. Li and B. Jacob, "Statistical DRAM modeling," in *Proc. International Symposium on Memory Systems (MEMSYS 2019)*, Washington DC, Oct. 2019, pp. 521–530.