

Review: “A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM” (1)

Summary

The paper outlines the issue of inefficient DRAM operation scheduling. It explains the issue and proposes several solutions. It simulates these and evaluates their performance. It does this in the following steps:

- The paper explains that a DRAM is built from different banks which intern are made of rows. To access a row in a bank first an ACTIVATE command is issued, then a READ/WRITE command is issued and finally a PRECHARGE before switching to another row. Each of these commands take time but they are not mutually exclusive but are treated as such in common DRAM chips.
- The paper proposes three solutions to parallelize the memory access and overlay the delays created by the different operations.
 - **SALP-1** is the first solution proposed. It overlays the PRECHARGE and ACTIVATE commands if the access is to two different subarrays. This is possible as both the PRECHARGE and the ACTIVATE command are local to their respective subarray. This does not require any changes to the DRAM chip. The only changes necessary are to the memory controller.
 - **SALP-2** is the second solution. It overlays the write recovery latency with a following ACTIVATE command. This is normally not possible as they both depend on a global latch holding the address to be written/read. By introducing an additional such latch in each subarray the overlay works and a speedup can be achieved.
 - **MASA** is the third proposed solution. It works by allowing several subarrays to be active at the same time with the memory controller designating one subarray at a time to drive the global bitline.
- The three proposed solutions are tested on a cycle-accurate DDR3-SDRAM simulator that was developed by the researchers. They compare the performance of their solutions against conventional systems on a broad set of tests and observe clear performance gains.

- The paper claims that these changes are low cost and low power.
- The paper compares its solutions to other solutions such as in memory cache or more banks in DRAM and concludes that their solution achieves a similar result as those and that these solutions could work in tandem. It further outlines that the solutions proposed in the paper are lower cost than the alternatives.

Strengths

The paper identifies an important issue and proposes three solutions with variable implementation cost. This is very commendable as while the MASA approach is clearly the strongest of the three it is also the most costly. But on the other hand the SALP-1 approach can be used with very low implementation cost. Mainly the SALP-1 and SALP-2 solutions are very smart uses of already existing logic and achieve a significant improvement with minimal changes. The paper also provides a fair comparison with other proposed solutions.

Weaknesses

The MASA approach is clearly very involved and while it could be used in a high performance environment it is unlikely to be broadly adopted. The cost estimation is hard to follow and oversimplified. Therefore it has to be taken with a grain of salt. The system was not tested in a physical implementation. The results are only simulated and therefore must be taken with a grain of salt as the real-world effects such as process variation temperature, etc. are hard to model accurately. Further the concrete impact on the yield of such DRAM chips is not really assessed in the paper.

My POV

I believe that the paper provides important approaches to improve DRAM performance. While I see great value in SALP-1 and SALP-2 I do not see MASA to be viable for wide spread application. It would be nice to see the performance of the solutions in physical implementations as it is hard to assess the results on a purely simulation basis.

Takeaways

- DRAM performance can and has to be improved. This can be done through brute force or by smartly optimizing the DRAM structure and timing.

- It is hard to assess the true performance of a system if only simulated results are provided as they tend to neglect real world effects.