

# 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 in turn 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 an 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.

# Review: “An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms” + Retrospective (2)

## Summary

### Paper

The paper discusses the problem with DRAM bitflip error profiling by highlighting data pattern dependence and variable retention times as issues that are overlooked by prior methods. It does this in the following steps:

- The paper establishes that past methods assumed that DRAM bitflip errors were neither data pattern dependent nor changing over time. It proposes a set of tests to test for these the hypothesis that there is indeed a data pattern dependence and that the retention times are variable. For this it uses the following test cases:
  - All 0s/1s to provide a static uniform test.
  - Checkerboard to provide a static non uniform test.
  - Walk to provide a continuous, changing, and predictable test.
  - Random to provide a continuous, changing, and unpredictable test that might catch overlooked cases and therefore provides a higher coverage.
- The paper first shows that their setup testing DDR3 chips from several manufacturers show an exponential temperature dependence as already shown in prior work to both confirm the prior work and validate their methodology as it provides a matching base case.
- The paper provides results that clearly show that there is both a data pattern dependence and a variation in retention times for DRAM. They also show that this dependence varies heavily from different technologies and manufacturers. The paper also shows that these errors occur more often in newer technology.
- The paper concludes that a more rigorous and integrated profiling process is needed to assess the retention times of DRAM-cells and optimize performance.

### Retrospective

The retrospective provides a context for the research provided in the paper and outlines the cooperation between the manufacturers and the researchers. It also provides several examples of research building on the papers results.

### Strengths

The paper clearly addresses an important issue and provides a fair investigation over a varied test set. The paper demonstrates validity of its methodology by comparing to prior results. The results had a strong impact on following research as it was an early exploration of the issue of more complex DRAM retention time profiling.

### Weaknesses

**The paper** has very few weaknesses and any I can come up with are only nitpicks. The paper could have tested on more DRAM chips the paper could have provided a more diverse test set, etc.

**The Retrospective** on the other hand does have clear weaknesses. While it provides context to the research presented in the paper and showcases the papers impact it is also very self-indulgent and does not add much to the research provided in the paper.

### My POV

The paper is very valuable research that dared to go a first step into a direction that had not previously been explored. It makes a very strong case and I do not have any concrete improvements on neither the paper nor its concepts. I do not see any tangible value in the retrospective.

## Takeaways

- DRAM-cell retention time is both data pattern dependent and is variable.
- DRAM-cell retention is an important issue that needs to be addressed in future research. And there is great value in doing so.

# Review: “RAIDR: Retention-Aware Intelligent DRAM Refresh” + Retrospective (3)

## Summary

### Paper

The paper outlines the issue posed by the differences in required refresh rates of DRAM-cells. It proposes RAIDR as a solution to this problem and provides a simulation of the RAIDR system. It does this in the following steps:

- The paper explains the fundamentals of DRAM and shows that not all cells require the same refresh rate. The paper therefore concludes that a lot of time and energy is wasted refreshing DRAM-cells that do not need to be refreshed.
- The paper proposes RAIDR as a solution to the refresh problem. The idea of RAIDR is to group cells into bins with different refresh rates. Each bin corresponds to a minimal refresh rate and all cells in that bin get refreshed with this refresh rate. The bin information is stored via bloom filters which allows implementation with minimal cost.
- With RAIDR there only needs to be an initial profiling and grouping of the cells into the bins and then the DRAM can run with these different refresh rates.
- The paper compares its approach to similar approaches and outlines strengths, weaknesses and interoperability.
- The paper provides simulation results for the RAIDR implementation which show that it provides significant speedup compared to the baseline. The results further show that the idle power consumption of RAIDR is lower than the baseline giving it a clear advantage for idle DRAM refreshes.

### Retrospective

The retrospective gives background on professor Mutlu’s work on DRAMs and work in the 2000s. It states that the main impact of the RAIDR paper was as inspiration for following research. The retrospective also admits that RAIDR does not account for VRT or DPD. Further the retrospective explains that the with the help of industry a physical testing setup could be realised after the publication of the original paper.

### Strengths

**The paper** works on the important topic of DRAM-cell refreshing and outlines a strong core idea of different refresh times. It provides an initial ansatz to solve this problem with RAIDR. The paper provides a thorough analysis of the system and its simulated performance. **The retrospective** gives important context for the paper and provides the existence of the open source physical implementation of the system.

### Weaknesses

The RAIDR system as proposed in the paper does not address DPD or VRT which is a clear weakness. Further it only relies on simulated results at least in its initial form but it is clarified in the retrospective that a physical implementation was built which seemingly matched the simulated performance - the results are not provided directly - of RAIDR.

### My POV

With the benefit of hindsight I would clearly account for DPD or VRT but this is an unfair perspective and must admit that the paper laid important groundwork in the area of DRAM refresh optimization. The fact that a physical implementation was realised is of great value to me as it builds a clear link between theory and application.

### Takeaways

- DRAM-cell refresh optimization is fundamental to further improve the performance of DRAM chips in the future. This paper provides a very early approach to solving this issue.

# Review: “MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Processing” (4)

## Summary

The paper realises that current PUD implementations use a SIMD and are therefore inefficient. It proposes a MIMD approach to DRAM they call MIMDRAM. The paper proposes a hardware/software-codesign which more efficiently utilize the PUD resources. The system is benchmarked and compared to similar systems and comparable baselines. This is done in the following steps:

### Problem:

- The paper outlines that current systems use a SIMD approach to PUD. In that approach a DRAM-line is used as a big vector on which an operation is performed. This can lead to underutilization if only a fraction of the line is filled with usable data.
- Another issue is that the data transfer and communication between different lines is very difficult. Therefore the systems are limited to “map”-style operations.
- The last problem the paper wants to tackle is that common PUD implementations demand that either the programmer or the compiler writers manually manage the mapping of the data, alignment, instruction scheduling.

### MIMDRAM System:

- The MIMDRAM system has a fine-grained DRAM segmentation which is realised by segmenting the global wordline into parts called DRAM mats.
- The system introduces data buses to move data within and across mats.
- MIMDRAM has its own control and scheduling unit to handle the computations in DRAM.
- The paper introduces an LLVM-based toolchain to allow direct compilation into a MIMDRAM operatable state.
- The paper proposes a new memory allocator to provide transparent mapping & allocation of memory for PUD.

### Results:

- The system is simulated and tested on 12 real-world benchmark applications and compared to state-of-the-art CPUs and GPUs as well as SIMD RAM.
- MIMDRAM provides a performance improvement for most workloads and a higher energy efficiency.
- Further an implementation in HDL and evaluation with Synopsis shows that the area cost is modest compared to systems with similar performance.

## Strengths

MIMDRAM improves on the SIMD RAM workflow by more efficiently utilizing space in the DRAM due to the partition of the DRAM into mats. Further it improves on basic Ambit functionality by allowing for more complex operations due to the added buses between the mats.

## Weaknesses

The system requires a high integration in the CPU by demanding a purpose built CPU to work with MIMDRAM like with a coprocessor. The implementation of the proposed segmentation and buses requires substantial changes to preexisting DRAM designs which makes fabrication more complicated and reduces yield as well as increasing price. The performance gains are dependent on the level of parallelizability in the programs with some programs performing worse than the baseline. The paper does not provide a physical implementation therefore any results have to be taken with a grain of salt.

## My POV

I believe that MIMDRAM is an important step in the development of a strong PUD environment. It takes bold steps and provides an attempt at integration of several PUD ideas into one coherent system. While I do not believe that MIMDRAM itself is a viable system I still believe that it goes in the right direction and is important research on which further research can build.

## Takeaways

- Fully integrating PUD ideas into a coherent systems by necessity incur high integration cost as well as interoperability costs.
- MIMDRAM provides an important perspective on what a fully functioning PUD system could look like and what requirements still have to be met for it to work.

## Review: “VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency” (5)

### Summary

The paper discusses the impact of the DRAM charge time during refresh operations on the performance of a DRAM. It provides a simulator the charging and leakage of DRAM-cells and it proposes a system to more efficiently recharge DRAM-cells called variable latency refresh DRAM.

- The paper shows that 40% of the recharge time when recharging a DRAM-cell is spent charging the last 5% of the cell. It supposes that if at least for some recharges a cell is only charged to 95% instead of 100% a significant speedup in the recharge operation can be achieved.
- The paper provides an open-source model to simulate the recharge latency.
- The paper introduces a model where for each cell the DRAM-controller keeps track of its last recharges and decides whether to fully or only partially recharge the cell. This model is designed to work within the volatile constraints given by process variation.

### Strengths

The paper makes a strong case for variable recharge rates and provides a high-value, open-source model to simulate this process. The model is shown to be accurate by comparison with SPICE which is state-of-the-art and it is shown to be faster than SPICE. The paper also proposes a solution for the problem it discusses by introducing their VRL-DRAM controller extension.

### Weaknesses

The provided solution is only a concept and is not tested on a wide set of parameters. While there is merit in further pursuing the direction, the paper only gives an initial impulse which is far from a real solution to the problem. The open-source solution is locked behind a contact wall on gitHub which does not make it not open source but limits the ease of access.

### My POV

The proposed VRL-DRAM model is a valuable tool the effectiveness of which I cannot assess. I find the proposed solution to be too far away from an actual implementation to fairly assess it and don't quite understand its place in this paper as the main contribution seems to be the VRL-DRAM model.

### Takeaways

- The recharge latency of DRAM-cells is an important parameter and the refresh times are variable given the correct methodology.
- The provided model is a valuable contribution to future research.

## 6 Review: “Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture” (6)

### Summary

The paper discusses the issue of DRAM speed as a major drag on overall performance of a computer system. It elaborates that the main source of DRAM latency is the bitline capacitance which is proportional to the bitline length. This issue is commonly addressed by DRAM producers as a tradeoff between speed and cost. The paper offers a solution taking advantage of both the speed of short bitlines and the cost of long bitlines.

- The paper explains the fundamental tradeoff in conventional DRAMs. A shorter bitline means a lower bitline capacitance which is easier to drive to a desired voltage. But a shorter bitline means that more bitlines are needed to provide the same amount of memory. As each bitline needs a sense-amplifier this increases the cost of the DRAM considerably. For a longer bitline the exact opposite is the case.
- The paper proposes a solution that utilizes a gate in the middle of the bitline to split it into two sections. A close and fast section and a far and slow section. Therefore allowing both the benefits of short and long bitlines in conventional designs at small cost.
- The paper establishes the issue of managing the now partitioned memory. The easiest but also worst performing option is to use it like normal DRAM that has a faster and a slower memory partition. But performance gain can be increased in two further ways.
  - The fast memory can be handled as a cache only visible to the DRAM for the slow memory allowing a better on average performance but decreasing the amount of memory on the DRAM. The paper proposes several caching strategies to optimally use the memory and maximise performance.
  - The fast memory can be made visible to OS allowing the OS to implement a tailored solution by deciding what data to store in the fast and what data to store in the slow memory.
- The paper evaluates the performance of the TL-DRAM by performing extensive simulations of the system.

### Strengths

The paper proposes a novel approach to DRAM implementation. The proposed system achieves very high performance on minimal cost, or at least area gain, on the DRAM chip. The paper very thoroughly simulates the system and provides a comprehensive analysis of the proposed implementation. The paper addresses an important issue in DRAM design.

### Weaknesses

The papers cost calculations are insufficient and biased. The paper assumes that the cost of the DRAM chip is linear in its size and then compares these metrics to the price of conventional DDR3-RAM. This is not a fair comparison as the cost of DDR3-RAM is lowered by its high production volumes. The TL-DRAM does not achieve the performance of fast DRAM chips and I do not see a valid space for this design on the market as it cannot realistically beat the cheap implementations in cost and does not outperform the fast implementations. The paper does not provide a physical implementation of the proposed system and therefore all results have to be taken with a grain of salt. To optimally use the TL-DRAM, i.e. showing it to the OS, the OS needs to be redesigned to be able to handle it which incurs small interoperability if it is not widely adopted.

### My POV

The issue addressed in the paper is an important one and the solution proposed is both innovative and effective but I cannot see this system being used in a commercial system in the near future. I find the cost estimation very biased and therefore disagree with the applicability claims of the paper. If the paper would have provided a physical implementation then the TL-DRAM could have been tested in a real context and the results would have more weight. Further the cost and viability could be assessed more fairly.

### Takeaways

- DRAM-latency is an important issue that needs to be addressed.

- The proposed TL-DRAM is a smart implementation using preexisting structures with minimal modifications to achieve remarkable results.
- The real-world applicability of this paper is flawed at best as it proposes a theoretical concept and assumes very optimistic cost estimations.