

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 approach 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 SIMDRAM.
- 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 SIMDRAM 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.