

The Role of Memories in Transprecision Computing

Christian Weis*, Matthias Jung†, Éder F. Zulian*, Chirag Sudarshan*, Deepak M. Mathew*, Norbert Wehn*

*Microelectronic Systems Design Research Group, University of Kaiserslautern, Germany

Email: {weis, zulian, sudarshan, deepak, wehn}@eit.uni-kl.de

†Fraunhofer IESE, Kaiserslautern, Germany

Email: Matthias.Jung@iese.fraunhofer.de

Abstract—Computing paradigms largely evolved over the last decades mainly driven by continuously increasing performance requirements, energy efficiency and power density challenges. Heterogeneous highly parallel architectures enhanced with dedicated accelerators tuned to specific applications, near-threshold computing, and recently approximate computing are examples of these new approaches. In this context the memory part was relatively untouched. However, memories play a central role in any computing system, are a major source of power consumption, and limit in many applications the overall compute performance. In this paper, we focus mainly on Dynamic Random Access Memories (DRAMs), which are today’s most prominent external memories. In transprecision computing we address the DRAM memory challenge by several new approaches that are strongly related to the new techniques known on the compute side. In particular, these are the concept of approximate DRAM, advanced power-down modes and the integration of application knowledge into the memory system.

I. INTRODUCTION

External memories are an indispensable part of every computing system. With the increasing performance of computing platforms and the exponential increase in produced data, memory bandwidth and capacity are moving more and more into the focus of computing. Today’s most prominent external memories in computing systems are *Dynamic Random Access Memories* (DRAM). DRAMs largely contribute to the overall power consumption and have a significant impact on the overall compute performance. Already today, many applications are not compute but memory bandwidth limited. E.g., Google has recently shown for its *Tensor Processing Unit* (TPU) ASIC [1] that four out of six neural network inference executions are not compute but memory bandwidth limited. In natural language processing more than 80% of the energy consumption comes from the DRAMs and only about 20% from the computing itself. The performance gap is mainly due to the difference in the business model driven goals between processing and DRAM memories: on the one hand processors are optimized for performance, DRAMs, on the other hand, are optimized for bit density, i.e. minimum cost/bit. Although the peak performance of DRAMs has largely improved over the years due to the increased internal pre-fetching and the faster interface performance, Fig. 1 shows the corresponding peak performance for different types of state-of-the-art DRAMs, the internal DRAM core frequency has not changed over time to keep cell sizes small. This can provoke up to 10x access time variations depending on the status of the banks, i.e. row/page misses. Thus, the real DRAM performance can be much lower

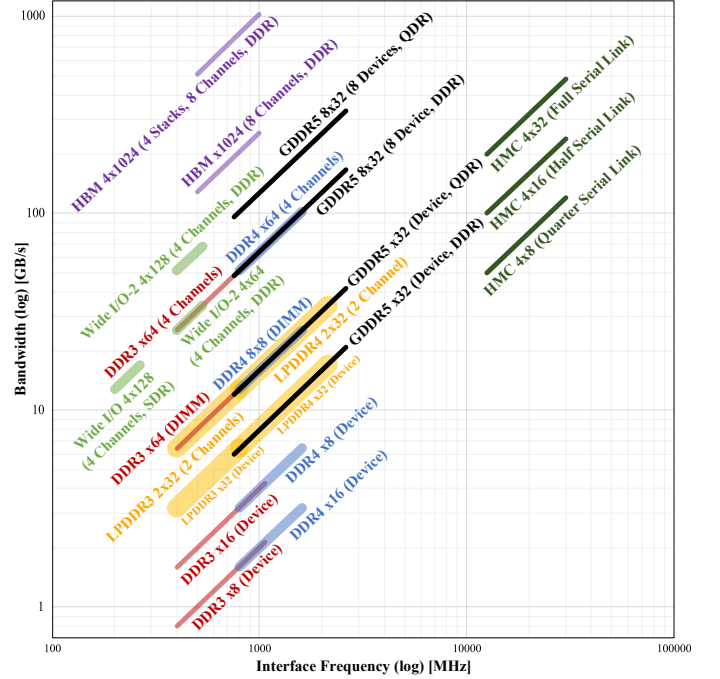


Fig. 1. I/O Frequency and Maximum Bandwidth of Different DRAM Types

than the peak performance. As we can see from Fig. 1, the different DRAM types largely vary in their peak performance due to their different internal architectures. This results also in a large difference in max memory capacity and energy/bit, see Fig. 2. This figure shows the existing trade-off between energy efficiency and maximum memory capacity. Another challenge in DRAMs is the refresh operation. DRAMs have to be periodically refreshed due to their dynamic nature. This refresh largely consumes energy and reduces the available memory bandwidth. E.g. a 4TB DDR3 DRAM memory system consumes ≈ 300 Watts only by refreshing the cells [2]. Advanced stacked DRAM architectures, such as Wide I/O DRAM or HBM (*High Bandwidth Memory*), exacerbate this penalty due to the exponential temperature dependency of the DRAM cell’s leakage that provokes an increased refresh rate.

The transprecision computing approach is a new computing paradigm that aims at demolishing the ultra-conservative precise computing abstraction by replacing it with a more flexible and efficient one. This means exploiting approximation in hardware and software from a statistical and a deterministic

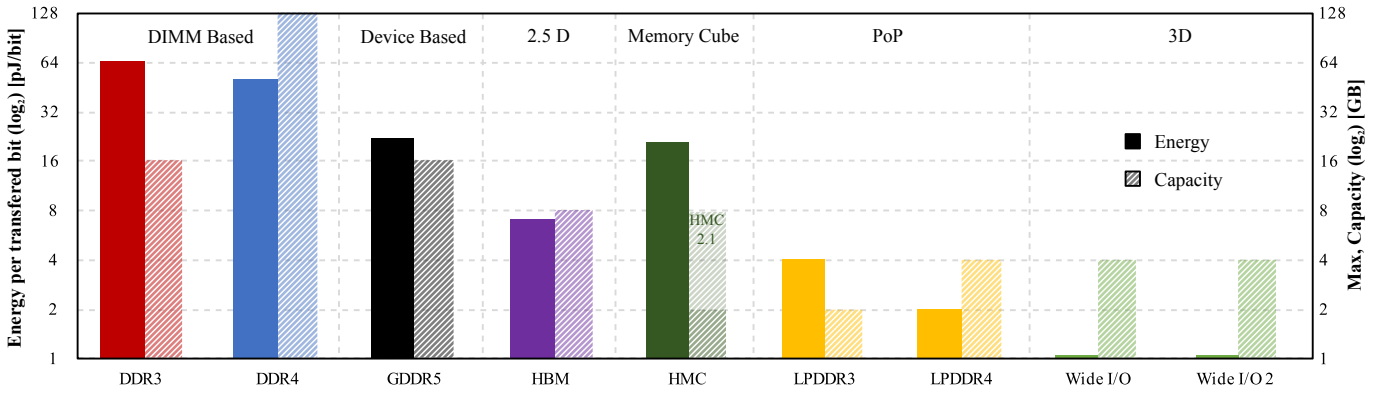


Fig. 2. Properties of Today's DRAMs (Sources: Micron, Hynix, Nvidia, Xilinx, JEDEC)

viewpoint. The main driver for this new computing paradigm is energy efficiency, ranging from tiny nodes for Internet-of-Things (mW platform) to HPC systems (kW platform). For more details see [3]. Thus, we address the DRAM memory challenge in transprecision computing by several new techniques. First, the concept of approximate computing is transferred to DRAMs [4]. The key knob in approximate DRAM is the refresh period. Here, we lower or even completely switch-off the refresh. Obviously, this can result in possible data errors [5] and is very application dependent.

We also exploit application knowledge to improve the energy efficiency and bandwidth by optimizing the memory controller. The DRAM memory controller orchestrates the DRAM accesses from the compute fabrics to the DRAMs themselves. It is a complex device that is, in contrast to today's compute architectures, still general purpose with limited context information. This context information is restricted to the buffer size. Here, we follow an *Application Specific Memory Controller* (ASMC) approach [6]. For a given application specific access pattern, the memory controller can be configured in such a way that the data are optimally mapped into the DRAM banks and rows. An optimal mapping minimizes the row misses and, hence, maximizes the bandwidth and energy efficiency. Furthermore, we use special power-down [2] and refresh strategies [7], which are also tuned to the application characteristic. We limit our optimizations not only to DRAMs but we will also consider emerging memories, such as Resistive RAMs (ReRAMs).

The remainder of the document is organized as follows, first we present briefly the concepts of approximate DRAMs and the application specific memory controller. Then we highlight a virtual platform systems to analyze and explore the memory architecture design space in transprecision computing systems. Finally we present the memory systems for the target platforms (demonstrators).

II. APPROXIMATE DRAM

In approximate DRAM we transfer the concept of approximate computing towards the memories. As mentioned above, by increasing the retention time we will accept a certain inherited error probability during the data lifetime in

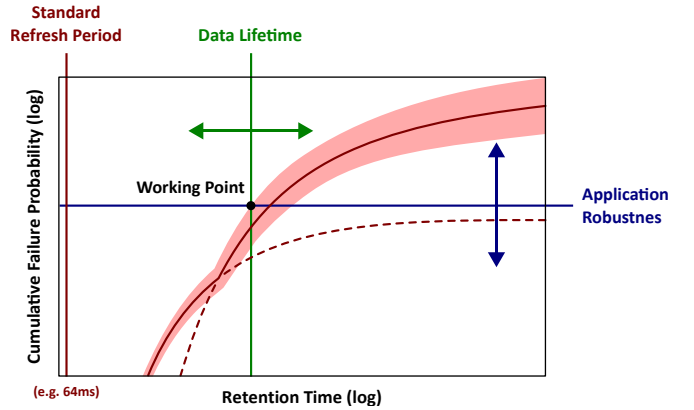


Fig. 3. Qualitative Retention Error Behavior

the memory system. This is similar to the voltage supply reduction of an on-chip working memory, such as SRAM. Thus, the error occurrence probability during write/read will increase. For DRAMs the major mechanism is the refresh rate reduction, i.e. increasing the retention time for the cells, due to the charge-based storage in contrast to the feedback-based storage in SRAMs. Fig. 3 shows the design space for using approximate DRAMs. This figure shows that there is a large potential for an increased retention time that translates into improvements in energy efficiency and bandwidth, dependent on the data life times and the application robustness. If the data lifetime is shorter than the actual required refresh rate in the DRAM, the refresh can consequently be disabled for the respective address range without having any impact on the failure behavior. One prominent example is actually the frame buffer of a graphics processing unit (GPU) that will be implicitly refreshed due to loading/writing of new updated data (assuming 40-50 frames per second). Thus, this frame buffer requires no explicit refreshing at all. With this approach we move away from the ultra-conservative retention time towards application dependent retention times.

A deep understanding of the retention behavior of DRAMs is mandatory to make efficient use of the approximate DRAM approach. The retention behavior strongly depends on the temperature. Beyond that there is the possibility to use error

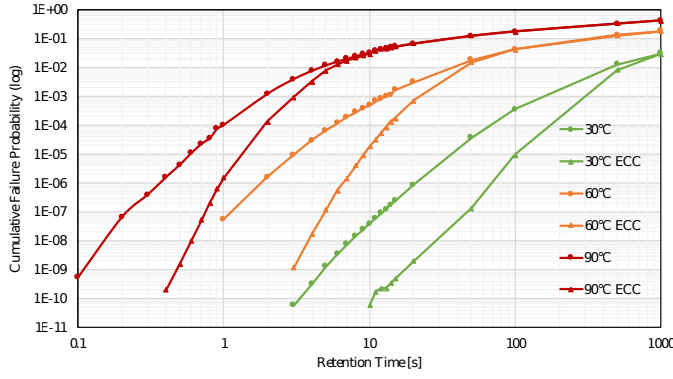


Fig. 4. Retention Time Measurements using Random Data of a Recent 4GB DDR4 SO-DIMM using the measurement platform [5]

correcting codes (ECC) to improve the error behavior. To the best of our knowledge, no in-depth investigations was performed so far to explore the trade-off between retention time behavior, temperature and ECC techniques for advanced DRAM devices. Fig. 4 demonstrates this trade-off for a 4GB DDR4 Small-Outline Dual-Inline Memory Module (SO-DIMM). With ECC, the failure probability at the lower end of the retention curve (< 1 s) can be reduced by *two to four* orders of magnitude even at 90°C . In further investigations we observed that, independent of the DRAM vendor, the typical (72,64) Hamming codes are not much effective in correcting the retention failures at higher temperatures ($> 60^\circ\text{C}$) and longer retention times (> 10 s).

III. APPLICATION SPECIFIC MEMORY CONTROLLER

As mentioned before, the achievable bandwidth and energy efficiency of DRAM memories strongly depends on the access pattern context that is application dependent. Many applications have a regular or fixed memory access pattern, either temporal (the commands follow a well-defined rule) or spatial (the sequence of the accessed addresses is deterministic). For more than a decade, inherent application specific knowledge has been heavily utilized by techniques, such as dedicated hardware accelerators or *Application Specific Instruction-Set Processors* to improve the overall energy efficiency. However, on the memory side, there is a limited amount of research that exploits such knowledge. In transprecision computing we pursue for the first time a customized application specific

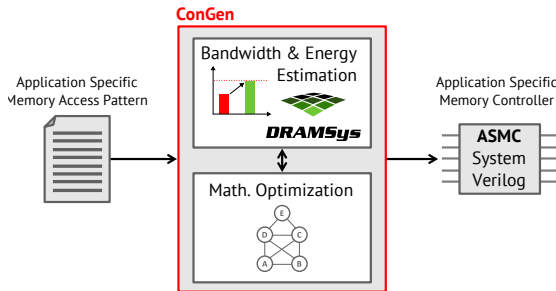


Fig. 5. ConGen methodology to generate an ASMC

memory controller approach, named ASMC [6]. In contrast to commercial off-the-shelf DRAM controllers, which are targeting general purpose systems, an ASMC is lean, highly energy and area efficient while largely improving the achievable bandwidth for the specific target applications.

Fig. 5 shows our methodology. The input is the application specific memory access pattern. Using discrete optimization techniques we can calculate an optimum mapping from logical to physical DRAM addresses (this mapping is called scrambling), such that the number of row hits is maximized. Maximizing row hits maximizes the bandwidth and minimizes the energy. The scrambling process fully exploits the bank level parallelism inside the DRAM. The scrambling information is stored in small Look-Up tables that control multiplexer banks. So the resulting front-end of the DDR memory controller has a low complexity. The ASMC approach outperforms general purpose memory controllers. Although these controllers use advanced online scheduling algorithms and quality of service methods to increase the memory bandwidth, they suffer on their local view due to the limited buffer depth.

IV. VIRTUAL PLATFORM (VP) BASED EXPLORATION FRAMEWORK

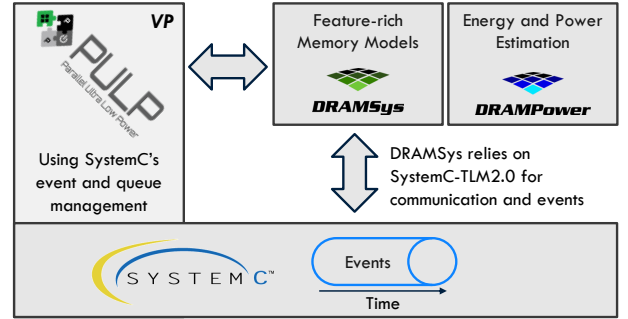


Fig. 6. Coupling of DRAMSys with the PULP [8] Virtual Platform for Exploration of Transprecision-capable Memories

The use of approximate DRAM, ASMC and advanced power down techniques in application context requires a corresponding platform for design space exploration. Such a platform must be very fast to allow real system behavior simulation, on the other hand, the memory models have to be quite accurate to reflect the real DRAM behavior such as accurate timing, energy and retention behavior. The virtual platform presented in [9], [10], [11] enables a fairly accurate modeling of performance, power consumption, and reliability of memory systems, named DRAMSys and DRAMPower. They support a wide range of standard and emerging DRAM subsystems such as DDR3, DDR4, LPDDR3 and Wide I/O memories. The framework is composed of flexible and extensible models and is based on SystemC-TLM as the main virtual platform infrastructure. It can be connected to any TLM2.0 based core and bus models for generating input data for memory subsystem, e.g., the platform can be seamless connected to gem5 [12] to enable full system simulation. In the context

of transprecision computing the platform is coupled to the *Parallel Ultra Low Power* (PULP) System [8] as depicted in Fig. 6 to run exhaustive explorations. PULP is an open-source many core platform based on energy-efficient, 4-stage, in-order RISC-V processor cores. Two demonstrators are envisaged, that are optimized for two different power envelopes (named kW and mW platform), and both of them are based on the PULP platform.

V. MEMORY SYSTEM FOR THE kW DEMONSTRATOR

The kW architecture serves as demonstrator to show the efficiency of the transprecision computing paradigm for HPC computing. Here, a Power8 architecture is connected via CAPI interface to an FPGA that hosts an array of PULP based processing systems (clusters) and the memory controller. The architectural structure is shown in Fig. 7.

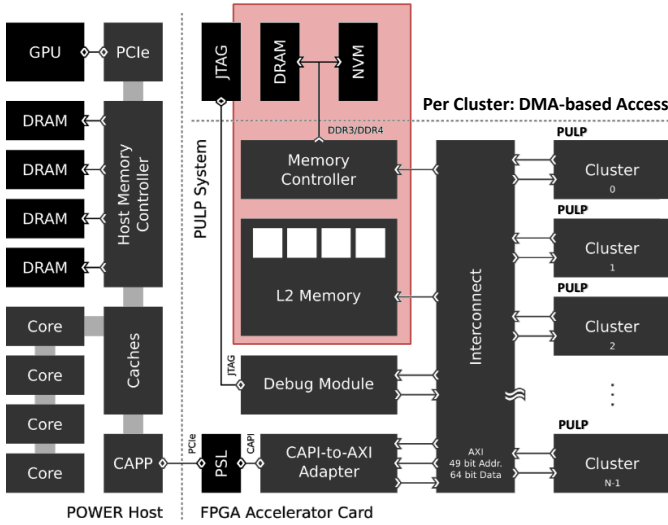


Fig. 7. Memory System for the kW Platform Architecture

The current memory controller supports DDR3 and DDR4 devices and includes enhanced features for multiple PULP clusters that access the DRAM via multiple DMAs (one DMA per cluster). The controller also supports application specific address mappings using the aforementioned *ConGen* methodology. Approximate DRAM is leveraged by sophisticated refresh techniques including detailed Row-by-Row refresh control with an optimized Row-Granular-Refresh (ORGR) [7]. Pro-active power-down strategies and ECC are integrated to further increase the energy efficiency of the memory subsystem. Current investigations put emphasize on the support of heterogeneous memory systems composed of DRAMs, and non-volatile Memories (NVMs), such as ReRAM.

VI. MEMORY SYSTEM FOR THE mW DEMONSTRATOR

As stated above, the mW demonstrator intends to show the advantage of transprecision computing for highly energy efficient compute nodes used in the context of IoT. Due to this fact and the low pin/pad count requirement, it is impossible to integrate a fully-fledge SO-DIMM DDR3/4

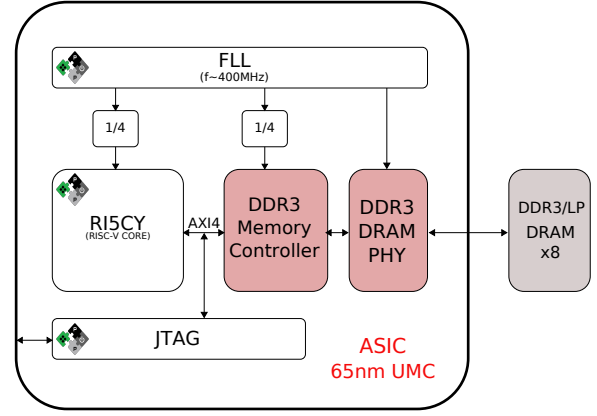


Fig. 8. Memory System for the mW Platform Architecture

DRAM channel with 64 data I/Os. Instead, a lean down-scaled DDR3 memory controller with a 4x RISC-V core PULP cluster is implemented (see Fig. 8). This system has a reduced data width (x8) and performance (maximum freq=533MHz). The memory controller is optimized for low complexity and includes a complete PHY and DDR front-end controller. In contrast to the kW demonstrator that is implemented in an FPGA, the mW demonstrator is fully implemented in UMC's low-leakage (LL/LP) 65nm CMOS technology. The front-end of the memory controller includes the same features as the controller for the kW demonstrator, i.e., application specific address mapping, sophisticated refresh techniques and advanced power-down modes to achieve maximum energy efficiency.

VII. CONCLUSION

The memory subsystem is an essential part in any computing system and imposes many challenges since it is a major source of power consumption and often limits the overall performance. To improve the overall energy efficiency we heavily exploit application knowledge when designing the memory subsystem. An efficient virtual platform supports the exploration of the overall memory system for specific applications. Approximate DRAM that is based on sophisticated refresh control, application specific address mapping and very efficient power-down modes guarantees minimized energy consumption. The efficiency of our approach will be demonstrated on two transprecision compute architectures, i.e., a kW and a mW node that target different applications and different power envelopes.

ACKNOWLEDGMENT

This work was partially funded by the German Research Foundation (DFG) by grant no. DFG WE2442/10-1. The project OPRECOMP acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the EU's Horizon 2020 research and innovation programme, under grant agreement no. EU 732631 (<http://www.oprecomp.eu>).

REFERENCES

- [1] N. P. Jouppi *et al.*, “In-Datacenter Performance Analysis of a Tensor Processing Unit,” in *Proceedings of ISCA '17*. New York, NY, USA: ACM, 2017, pp. 1–12.
- [2] R. Jagtap, M. Jung, W. Elsasser, C. Weis, A. Hansson, and N. Wehn, “Integrating DRAM Power-down Modes in Gem5 and Quantifying Their Impact,” in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '17. NY, USA: ACM, 2017, pp. 86–95.
- [3] C. Malossi *et al.*, “The Transprecision Computing Paradigm: Concept, Design, and Applications,” in *Accepted for publication in Proceedings of the IEEE DATE Conference*, Dresden, Germany, March 2018.
- [4] M. Jung, D. M. Mathew, C. Weis, and N. Wehn, “Invited: Approximate computing with partially unreliable dynamic random access memory: Approximate DRAM,” in *ACM/EDAC/IEEE DAC*, June 2016, pp. 1–4.
- [5] M. Jung, D. M. Mathew, C. C. Rheinlnder, C. Weis, and N. Wehn, “A Platform to Analyze DDR3 DRAM’s Power and Retention Time,” *IEEE Design Test*, vol. 34, no. 4, pp. 52–59, Aug 2017.
- [6] M. Jung *et al.*, “ConGen: An Application Specific DRAM Memory Controller Generator,” in *International Symposium on Memory Systems (MEMSYS 2016)*, 2016.
- [7] D. M. Mathew *et al.*, “Using Run-time Reverse-engineering to Optimize DRAM Refresh,” in *Proceedings of the International Symposium on Memory Systems (MEMSYS)*. NY, USA: ACM, 2017, pp. 115–124.
- [8] D. Rossi *et al.*, “A 60 GOPS/W, -1.8 V to 0.9 V Body Bias ULP Cluster in 28nm UTBB FD-SOI Technology,” *Solid-State Electronics*, 2016.
- [9] M. Jung, C. Weis, and N. Wehn, “DRAMSys: A flexible DRAM Subsystem Design Space Exploration Framework,” *IPSI Transactions on System LSI Design Methodology (T-SLDM)*, August 2015.
- [10] K. Chandrasekar *et al.*, “DRAMPower: Open-source DRAM power & energy estimation tool,” <http://www.drampower.info>, <http://www.drampower.info/>.
- [11] C. Weis *et al.*, “Retention Time Measurements and Modelling of Bit Error Rates of WIDE I/O DRAM in MPSoCs,” in *Proceedings of the IEEE DATE Conference*, 2015.
- [12] C. Menard, M. Jung, J. Castrillon, and N. Wehn, “System Simulation with gem5 and SystemC: The Keystone for Full Interoperability,” in *Proceedings of the IEEE International Conference on Embedded Computer Systems Architectures Modeling and Simulation (SAMOS)*, July 2017.