

# Efficient Reliability Management in SoCs - An Approximate DRAM Perspective

Matthias Jung, Deepak M. Mathew, Christian Weis, Norbert Wehn  
Microelectronic Systems Design Research Group  
University of Kaiserslautern  
{jungma,deepak,weis,wehn}@eit.uni-kl.de

**Abstract**— In today’s computing systems *Dynamic Random Access Memories* (DRAMs) have a large influence on performance and contribute significantly to the total power consumption. Thus, recent research activities bring the idea of *approximate DRAM* into focus to save power and improve performance by lowering the refresh rate or disabling refresh completely. Hence, fast and accurate models are required for a thoroughly exploration of approximate DRAM for error resilient applications. In this paper we present a holistic simulation environment for investigations on approximate DRAM and show the impact on error resilient applications.

## I. INTRODUCTION

In the past, *Approximate* and *Probabilistic Computing* evolved as design paradigms that exploit the error resilience of applications to increase their performance and decrease their power consumption [11].

More recently these research activities have been extended from pure computing cores to uncore components, such as building blocks of communication systems [10] or the DRAM controller [7] and further the DRAM itself, often denoted as *Approximate DRAM* [23, 24, 1, 16, 27]. The underlying motivation for an approximate DRAM is the increasing power consumption and performance penalty caused by unavoidable DRAM refresh commands. Liu et al. [22] predicted that 40% to 50% of the power consumption of future DRAM devices will be caused by refresh commands. Moreover, 3D integrated DRAMs like Wide I/O or HMC worsen the scenario with respect to increased cell leakage, due to the much higher temperature. Therefore, the refresh frequency needs to be increased accordingly to avoid retention errors [28].

Approximate DRAM tackles these issues by lowering the refresh frequency (reducing the guard bands proposed by the vendors) or even disable the refresh completely and accepting the risk of data errors. Many applications have an inherent error resilience that tolerates these errors and therefore refresh power often can be reduced with a minimal loss of the output quality.

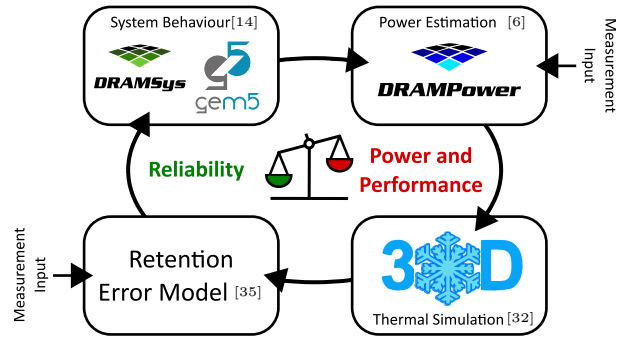


Fig. 1.: Simulation Framework for Approximate DRAM Explorations and Reliability Management in SoCs

The major contributions of this work are:

- We present a holistic simulation framework for exploration of efficient reliability management in System on Chips (SoCs) with focus on approximate DRAM (See Figure 1 and Section III).
- We show with measurements on Wide I/O and DDR3 devices that even at high temperatures DRAM can retain its content much longer than assumed in current literature. The simulation framework is calibrated with these measurements (Section IV).
- We show with the simulation framework the practical feasibility for approximate DRAM on two error resilient applications (Section V).

## II. RELATED WORK

### A. Approximate DRAM - Reliability vs. Power

Liu et al. presented the first work on approximate DRAM, called *Flicker* [23], which reduces the number of refreshes by partitioning a DRAM bank in a critical and non-critical region. The non-critical region will be refreshed with a lower refresh rate.

A similar approach is followed by *Quality Aware Approximate DRAM* [27], which characterizes the used

DRAM by extensive retention time measurements. As a result the DRAM pages are sorted into quality bins. During allocation critical data is stored in high quality bins, whereas approximate data is allocated in low quality bins. The whole DRAM is then refreshed with the same refresh rate, which makes this approach applicable to today's DRAM devices, since no changes of the internal DRAM structure are required. However, this approach is very time consuming due to the prior characterization and it is very challenging because of the *Variable Retention Time* (VRT) [17, 21] phenomenon. Moreover, there is an overhead to store the essential information to apply this technique (sorted page order).

The *REVA* [1] refresh scheme can be used in dedicated video applications. It refreshes only the important *region of interest* (ROI) in a video frame.

*Sparkk* [24] proposes the idea of permutation of the data bits on several DRAM chips that are refreshed with different rates. The most significant bits of a byte are located in a highly refreshed DRAM device and the least significant bits are stored in a less refreshed chip.

*Omitting Refresh* (OR) [16] shows that for dedicated applications refresh can be disabled completely without or with negligible impact on the application performance. This is possible if it is assured that either the lifetime of the data is shorter than the currently required DRAM refresh period or if the application is error resilient and can tolerate bit errors to some degree in a given time window. It is shown that especially for 3D-integrated systems with Wide I/O DRAM this strategy is beneficial. Similar to *Flicker* the memory is divided in an *unreliable* and *reliable* region, as depicted in Figure 2. It shows a simulation step of a 3D-integrated mobile MPSoC with a 4-channel Wide I/O DRAM and heterogeneous computing components on the logic die. The lowest 3D-DRAM layer has the highest average temperature. Hence, this layer requires a higher refresh rate than the rest of the DRAM stack [28]. Consequently, the lowest layer is a perfect candidate for applying OR, by tolerating an *unreliable* memory layer in order to save refresh power. While the upper *reliable* part of the stack is refreshed the *unreliable* region can be accessed exclusively, which is managed by the DRAM controller.

The authors of [29] and [31] introduce programming techniques to exploit unreliable memories by distinguishing reliable and unreliable data types.

### B. Retention Aware Refresh

There are several works on retention aware refresh, that also try to reduce the number of refresh events but unlike approximate DRAM they assure full data integrity [33, 22, 2, 20, 36, 34]. These techniques can be applied in fields of non-resilient applications. However, it is very difficult to characterize DRAMs, as they experience VRTs and *Data Pattern Dependencies* (DPD) [21, 35] for their retention times. Moreover, in [35] it is shown

that the temperature has a strong effect on VRT. Hence, it is infeasible during startup of a system to determine an exact list of weak cells that considers all parameters, such as temperature, retention time and DPD. A more recent idea, *AVATAR* [26], tries to overcome VRT issues by combining an online *Error-Correcting Code* (ECC) mechanism with row selective refresh.

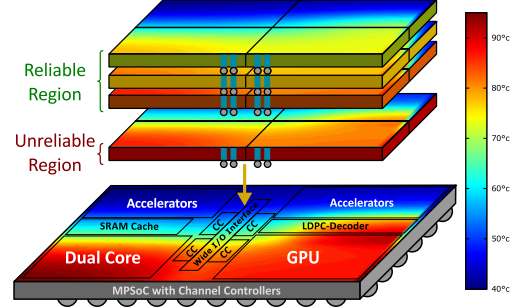


Fig. 2.: Wide I/O DRAM on an MPSoC [14], Simulated with our Approximate DRAM Framework

## III. SIMULATION FRAMEWORK FOR APPROX. DRAM

Modeling an approximate DRAM system is not trivial, since a reasonable tradeoff between accuracy and system simulation performance must be found. Therefore, we rely on SystemC Transaction Level Models (TLM) for fast and accurate simulation.

Figure 1 shows our closed loop simulation flow for investigations on approximate DRAM. This simulation loop is based on the modular DRAMSys framework [14] and consists of four key components:

- **DRAM and Core Models:** The DRAM model of the framework is based on a DRAM specific TLM protocol called *DRAM-AT* [15]. Due to TLM's modular fashion several types of DRAM and controller configurations can be modeled. For modeling the cores the *gem5* simulator is used [3]. We developed a coupling between *gem5* and SystemC to be able to integrate this powerful research platform in our simulation loop [13].
- **DRAM Power Model:** Since DRAMs contribute significantly to the power consumption of today's systems [19, 9], there is a need for accurate power modelling. For our framework we use *DRAMPower* [6, 5], which uses either parameters from datasheets, estimated via *DRAMSpec* [25] or measurements to model DRAM power. We modified this tool that it can be used as a library, which can be easily integrated in a C++/SystemC based simulator like our simulation framework to calculate the power consumption online during the simulation.
- **Thermal Model:** 3D packaging of systems like Wide I/O DRAM starts to break down the memory and bandwidth walls. However, this comes at the price

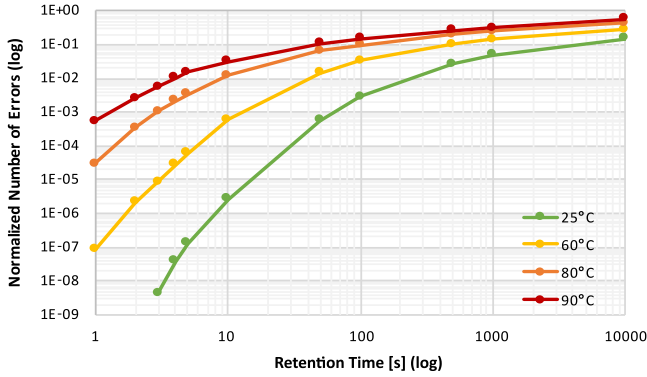


Fig. 3.: Measurements on DDR3

of increased power density and less horizontal heat removal capability of the thinned dies. The thermal issues of 3D ICs cannot be solved by tweaking the technology and circuits alone. It is crucial to analyze the behaviour of the whole system. Therefore, we integrated the thermal simulator 3D-ICE [32] in a SystemC wrapper [12] that is included in our closed-loop simulation for approximate DRAM analysis.

- **DRAM Retention Error Model:** this model is key to analyze the impact of approximate DRAM on executed applications. To study these energy vs. reliability trade-offs we developed a DRAM retention time error model [35] based on exhaustive measurements on Wide I/O and DDR3 DRAMs (Compare Section IV).

#### IV. MEASUREMENTS

To calibrate the models presented above we did exhaustive measurements [35] and estimated the retention times of Wide I/O [8, 17] and DDR3 DRAM devices [30] using different data pattern. We observed DPD and VRT, similar to the study in [21].

For this work we extended these measurements with respect to long retention times. The most research papers (e.g. [24, 22, 20]) rely on the study of Kim and Lee [18], where the maximum retention time of a DRAM cell is approximately 1000 seconds. As shown in the Figures 3 and 4 we observe that the majority of cells can hold data much longer than 10000 seconds for a 0xFF data pattern, even at high temperatures.

We see for the Wide I/O device after 10000 seconds only 6% of the cells are flipped at a temperature of 40°C while 24% are flipped for 100°C. For the DDR3 Device after 10000 seconds only 15% of the cells are flipped at room temperature (25°C) while 57% are flipped for 90°C. In contrast to common belief these measurements show that many DRAM cells can hold their value up to two orders of magnitude longer than assumed in [18]. We plan further measurements in the future with different DRAM devices from several vendors focused on process variation and VRT to enhance our simulation framework.

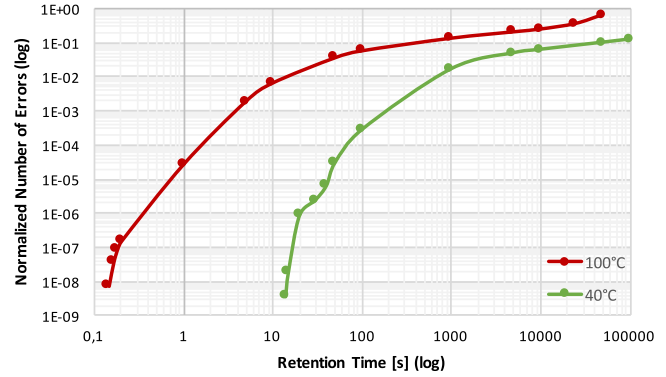


Fig. 4.: Measurements on Wide I/O

#### V. EXPERIMENTAL RESULTS

In a detailed case study [16] we used the presented simulation framework (Figure 1) to investigate the influence of an approximate DRAM with disabled refresh on two error resilient applications. For both following applications we modeled the architecture as shown in Figure 2:

- **Graph Processing:** Complex graphs are the heart of today's big data challenges e.g. recommendation systems used in social networks, content sharing services or online shops. The task for a good recommendation system is to find similarities in large graphs, which can be done by a fixed degree sequence model using the metric of co-occurrence [37]. In [4] an efficient hardware architecture is presented for calculating similarities in large graphs using the algorithm of [37]. We demonstrated that switching-off the refresh during co-occurrence calculation in graph processing has a negligible impact on the resulting quality. Since the memory is the bottleneck for this type of applications the available bandwidth can be increased by omitting refresh.
- **Channel Decoding:** In digital communication systems channel coding plays an integral role in all important modern communication standards. One of the currently most sophisticated concepts of channel coding is Low-Density Parity Check (LDPC) coding, which is part of recent wireless communication standards like WiGig and WiFi. Gimpler-Dumont et al. [10] have shown that channel decoding has an inherent error resilience. Therefore, we switched off refresh for this simulation on the input data buffer of an LDPC decoder and showed that the influence of the retention errors in the DRAM on the frame error rate after decoding is negligible compared to the influence of the errors induced by the communication channel.

For both applications described above the complete refresh energy can be saved. This results e.g. in 12% decrease of the total energy consumption of the channel decoding application.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a holistic simulation framework for investigations on approximate DRAM. The components in the framework are calibrated to conducted measurements. We showed that DRAM can hold its data much longer than commonly believed. In the future, we will extend our measurements, modeling and simulations to more DRAMs from several vendors with focus on technology, temperature, process variation, VRT and ECC mechanisms.

## ACKNOWLEDGEMENTS

The authors thank CEA-Leti and the involved students Peter Ehses, Éder Zulian and Martin Schultheis for the great support. This work was partially funded by the German Research Foundation (DFG) as part of the priority program Dependable Embedded Systems SPP 1500 (<http://spp1500.itec.kit.edu>), the DFG grant no. WE2442/10-1 and the Carl-Zeiss Stiftung.

## REFERENCES

- [1] S. Advani, N. Chandramoorthy, K. Swaminathan, K. Irick, Y. Cho, J. Sampson, and V. Narayanan. Refresh Enabled Video Analytics (REVA): Implications on power and performance of DRAM supported embedded visual systems. In *Computer Design (ICCD), 2014 32nd IEEE International Conference on*, pages 501–504, Oct 2014.
- [2] S. Baek, S. Cho, and R. Melhem. Refresh now and then. *Computers, IEEE Transactions on*, 63(12):3114–3126, 2014.
- [3] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.
- [4] C. Brugger, V. Grigorovici, M. Jung, C. Weis, C. D. Schryver, K. A. Zweig, and N. Wehn. A Custom Computing System for Finding Similarities in Complex Networks. In *Proceedings of the 2015 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 262–267, Montpellier, France, July 2015.
- [5] K. Chandrasekar, B. Akesson, and K. Goossens. Improved Power Modeling of DDR SDRAMs. In *proc. DSD’11*, 2011.
- [6] K. Chandrasekar, C. Weis, Y. Li, B. Akesson, O. Naji, M. Jung, N. Wehn, and K. Goossens. DRAMPower: Open-source DRAM power & energy estimation tool.
- [7] H. Cho, C. Cher, T. Shepherd, and S. Mitra. Understanding Soft Errors in Uncore Components. *CoRR*, abs/1504.01381, 2015.
- [8] D. Dutoit, C. Bernard, S. Cheramy, F. Clermidy, Y. Thonnart, P. Vivet, C. Freund, V. Guerin, S. Guilhot, S. Lecomte, et al. A 0.9 pJ/bit, 12.8 GByte/s WideIO memory interface in a 3D-IC NoC-based MPSoC. In *VLSI Technology (VLSIT), 2013 Symposium on*, pages C22–C23. IEEE, 2013.
- [9] N. Farahini, A. Hemani, A. Lansner, F. Clermidy, and C. Svensson. A scalable custom simulation machine for the Bayesian Confidence Propagation Neural Network model of the brain. In *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, pages 578–585, Jan 2014.
- [10] C. Gimmler-Dumont and N. Wehn. A Cross-Layer Reliability Design Methodology for Efficient, Dependable Wireless Receivers. *ACM Transactions on Embedded Computing Systems*, 2013.
- [11] J. Han and M. Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In *Test Symposium (ETS), 2013 18th IEEE European*, pages 1–6, May 2013.
- [12] M. Jung. IceWrapper - A SystemC Wrapper for 3D-ICE. <http://www.uni-kl.de/3d-dram/tools/icewrapper/>, October 2015.
- [13] M. Jung and N. Wehn. Coupling gem5 with SystemC TLM 2.0 Virtual Platforms. In *gem5 User Workshop, International Symposium on Computer Architecture (ISCA)*, Portland, OR, USA., June 2015.
- [14] M. Jung, C. Weis, and N. Wehn. DRAMSys: A flexible DRAM Subsystem Design Space Exploration Framework. *IPSJ Transactions on System LSI Design Methodology (T-SLDM)*, August 2015.
- [15] M. Jung, C. Weis, N. Wehn, and K. Chandrasekar. TLM modelling of 3D stacked wide I/O DRAM subsystems: a virtual platform for memory controller design space exploration. In *Proceedings of the 2013 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools, RAPIDO ’13*, pages 5:1–5:6, New York, NY, USA, 2013. ACM.
- [16] M. Jung, E. Zulian, D. Mathew, M. Herrmann, C. Brugger, C. Weis, and N. Wehn. Omitting Refresh - A Case Study for Commodity and Wide I/O DRAMs. In *1st International Symposium on Memory Systems (MEMSYS 2015)*, Washington, DC, USA, October 2015.
- [17] J.-S. Kim, C. S. Oh, H. Lee, D. Lee, H.-R. Hwang, S. Hwang, B. Na, J. Moon, J.-G. Kim, H. Park, J.-W. Ryu, K. Park, S.-K. Kang, S.-Y. Kim, H. Kim, J.-M.

- Bang, H. Cho, M. Jang, C. Han, J.-B. Lee, K. Kyung, J.-S. Choi, and Y.-H. Jun. A 1.2V 12.8GB/s 2Gb mobile Wide-I/O DRAM with 4x128 I/Os using TSV-based stacking. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 496–498, Feb 2011.
- [18] K. Kim and J. Lee. A New Investigation of Data Retention Time in Truly Nanoscaled DRAMs. *Electron Device Letters, IEEE*, 30(8):846–848, Aug 2009.
- [19] J. Krueger, D. Donofrio, J. Shalf, M. Mohiyuddin, S. Williams, L. Oliker, and F.-J. Pfreundt. Hardware/software co-design for energy-efficient seismic modeling. In *Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12, 2011.
- [20] C.-H. Lin, D.-Y. Shen, Y.-J. Chen, C.-L. Yang, and M. Wang. SECRET: Selective error correction for refresh energy reduction in DRAMs. In *Computer Design (ICCD), 2012 IEEE 30th International Conference on*, pages 67–74, Sept 2012.
- [21] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu. An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms. *SIGARCH Comput. Archit. News*, 41(3):60–71, June 2013.
- [22] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu. RAIDR: Retention-Aware Intelligent DRAM Refresh. In *Proceedings of the 39th Annual International Symposium on Computer Architecture, ISCA '12*, pages 1–12, Washington, DC, USA, 2012. IEEE Computer Society.
- [23] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn. Flicker: Saving DRAM Refresh-power Through Critical Data Partitioning. *SIGPLAN Not.*, 46(3):213–224, Mar. 2011.
- [24] J. Lucas, M. Alvarez-Mesa, M. Andersch, and B. Juurlink. Sparkk: Quality-Scalable Approximate Storage in DRAM. In *The Memory Forum*, June 2014.
- [25] O. Naji, C. Weis, M. Jung, N. Wehn, and A. Hansson. A High-Level DRAM Timing, Power and Area Exploration Tool. In *Embedded Computer Systems Architectures Modeling and Simulation (SAMOS)*, July 2015.
- [26] M. K. Qureshi, D.-H. Kim, S. Khan, P. J. Nair, and O. Mutlu. AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems. *Memory*, 2(4Gb):20, 2015.
- [27] A. Raha, H. Jayakumar, S. Sutar, and V. Raghunathan. Quality-aware Data Allocation in Approximate DRAM. In *Proceedings of the 2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, CASES '15*, pages 89–98, Piscataway, NJ, USA, 2015. IEEE Press.
- [28] M. Sadri, M. Jung, C. Weis, N. Wehn, and L. Benini. Energy Optimization in 3D MPSoCs with Wide-I/O DRAM Using Temperature Variation Aware Bank-Wise Refresh. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–4, March 2014.
- [29] A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze, and D. Grossman. EnerJ: Approximate Data Types for Safe and General Low-power Computation. In *Proceedings of the 32Nd ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '11*, pages 164–174, New York, NY, USA, 2011. ACM.
- [30] Samsung. M471B2874EH1-CF8 DDR3 1GB 533MHz, 2009.
- [31] F. Schmoll, A. Heinig, P. Marwedel, and M. Engel. Improving the Fault Resilience of an H.264 Decoder Using Static Analysis Methods. *ACM Trans. Embed. Comput. Syst.*, 13(1s):31:1–31:27, Dec. 2013.
- [32] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza. 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In *Proc. of ICCAD 2010*, 2010.
- [33] R. Venkatesan, S. Herr, and E. Rotenberg. Retention-aware placement in DRAM (RAPID): software methods for quasi-non-volatile DRAM. In *Proc. of HPCA*, 2006.
- [34] J. Wang, X. Dong, and Y. Xie. ProactiveDRAM: A DRAM-initiated retention management scheme. In *Computer Design (ICCD), 2014 32nd IEEE International Conference on*, pages 22–27, Oct 2014.
- [35] C. Weis, M. Jung, P. Ehses, C. Santos, P. Vivet, S. Goossens, M. Koedam, and N. Wehn. Retention Time Measurements and Modelling of Bit Error Rates of WIDE I/O DRAM in MPSoCs. In *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2015.
- [36] D. Zhu, R. Wang, Y. Wei, and D. Qian. Reducing DRAM refreshing in an error correction manner. *Science China Information Sciences*, pages 1–14, 2015.
- [37] K. A. Zweig and M. Kaufmann. A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining*, 1(3):187–218, 2011.