Improving the Error Behavior of DRAM by Exploiting its Z-Channel Property

Kira Kraft, Chirag Sudarshan, Deepak M. Mathew,
Christian Weis, Norbert Wehn
University of Kaiserslautern
Kaiserslautern, Germany 67663
{kraft,sudarshan,deepak,weis,wehn}@eit.uni-kl.de

Matthias Jung
Fraunhofer Institute for Experimental
Software Engineering (IESE)
Kaiserslautern, Germany 67663
matthias.jung@iese.fraunhofer.de

***Abstract*—In this paper, we present a new communication theoretic channel model for *Dynamic Random Access Memory* (DRAM) retention errors, that relies on the fully asymmetric retention error behavior of DRAM cells. This new model shows that the traditional approach is over pessimistic and we confirm this with real measurements of DDR3 and DDR4 DRAM devices. Together with an exploitation of the vendor specific true- and anti-cell structure, a low complexity bit-flipping approach is presented, that can largely increase DRAM's reliability with minimum overhead.**

I. Introduction

In this paper we show that DRAM's retention errors have an asymmetric behavior and how this asymmetry can be exploited to increase DRAM's reliability. Our proposed method is based on the observation that only a physical `1` in the DRAM can leak to a physical `0` ($1 \rightarrow 0$) and not the other way around ($0 \not\rightarrow 1$). The communication theoretic equivalence to this behavior is the *Z-channel*, in which only ones can flip to zeros with some fixed bit-error probablity. Unfortunately, viewing the DRAM from the outside, this error behavior is concealed by the vendor specific way of storing data in the DRAM, where some cells store the data as is, so a logical `1` is stored as a physical `1` (*true-cells*), and other cells store the data inverted, so a logical `1` is stored as physical `0` (*anti-cells*). With reverse-engineering, we reveal how logical bit values are stored as physical bit values, which paves the way to establishing the Z-channel as suitable DRAM retention error model. Knowing the retention error behavior of DRAMs is on the one hand valuable to explore new *Error Checking and Correction* (ECC) schemes that increase DRAM's reliability, and can help on the other hand to pursue approaches like *Approximate DRAM* (ADRAM), that has been presented in recent years [1]. With the technique presented in this paper we increase DRAM's reliability, especially for ADRAM, and show that the probability of retention errors can be reduced significantly by simple changes in the memory controller. In summary, the paper makes the following new contributions:

- We show that true- and anti-cells play a major role on the effect of *Data Pattern Dependency* (DPD) and that they have a huge impact on DRAM error behavior.
- We present a method to reverse-engineer the logical positions of true- and anti-cells and reveal the internal true- and anti-cell architecture for all major DRAM vendors. With this method every DRAM can be transformed logically into an entire true-cell DRAM.
- Knowing the internal structure of true- and anti-cells, we show that the error behavior of DRAMs can be modeled by the *binary asymmetric channel*, called *Z-channel*. This model minimizes the deviation from the real measured DRAM retention error behavior compared to other communication theoretic channel models.
- We present a new and low-overhead flipping scheme that reduces retention errors in ADRAM and thus increases its reliability.

II. Related Work

A. Approximate DRAM (ADRAM)

The underlying motivation of ADRAM is to reduce the increasing power consumption and increase the memory bandwidth by lowering the refresh frequency or even disabling the refresh completely while accepting the risk of data errors. Liu et al. present the first work on ADRAM, called *Flikker* [2], 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, so retention errors may occur in this region. Recently, further research on ADRAM has been presented in [3], [4], [5], and [6].

B. Cold Boot Attacks and Reverse Engineering of DRAMs

During their work on *Cold Boot Attacks*, where a DRAM DIMM is frozen and removed from the motherboard in order to restore private RSA encryption keys from it, the authors of [7] observed that for a different regions retention errors only happen from $0 \rightarrow 1$ and for other regions these error only happen from $1 \rightarrow 0$. The reason for their observation (true- and anti-cells) is explained in Section III. Paterson et al. [8] use the asymmetric behavior for a better reconstruction of RSA private keys. Moreover, recent works like [9], [10], and [11] show that internal information of DRAMs and their controllers can be reverse-engineered by sophisticated test setups. We will detail a similar technique in Section IV.

C. ECC for DRAM

A state-of-the-art ECC DRAM *Dual Inline Memory Module* (DIMM), for instance, consists of 8 DRAM devices and a 9th device, used to store the ECC redundancy. The code that performs the error correction and detection is a Hamming code, that is a *Single Error Correcting / Double Error Detecting*

978-3-9819263-0-9/DATE18/©2018 EDAA

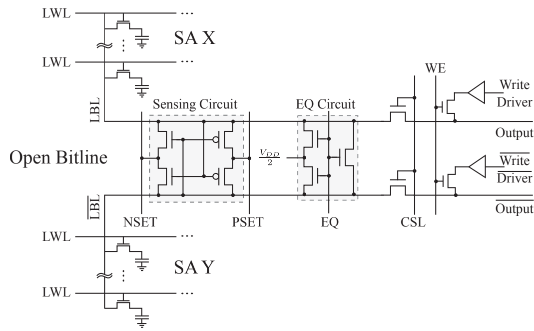


Figure 1: DRAM Subarrays and Sense Amplifiers

(SEC/DED) code [12]. Two vendors recently introduced on-die ECC [13], [14] to avoid retention errors. They lower the refresh rates by a factor of $4\times$ in order to reduce power consumption. This shows the importance of ECC for retention errors in DRAMs.

In addition to the above approaches, several other works motivate the use of Berger codes [15], that are only able to detect, but not correct, **all** asymmetric errors in a unidirectional channel (see Section V) [16], [17]. However, error detection is only helpful in cases where a retransmission of data is possible, which is usually not the case for ECC in memories.

III. DRAM Background

In the following we present the DRAM device architecture and functionality to understand how information is stored in DRAM cells. A single memory cell is built as a transistor capacitor pair where the data is stored in the capacitor as a charge – *noting that this charge can only leak out but not in again*. The main causes for the cell leakage current are the *PN-Junction Leakage* and *Trap Assisted Tunneling Gate Induced Drain Leakage* (TAT-GIDL) [18]. Therefore, every cell in the DRAM must be refreshed every $t_{REF} = 64\,\text{ms}$ in order to prevent so called *Retention Errors*.

The single cells are organized in *Sub-Arrays* (SA) which are connected to a sense amplifier as depicted in Fig. 1. In this example an open bitline configuration is used, which connects the *Local Bitlines* (LBL) to neighbored SA, in particular SA X and SA Y. It is important to know that only one bitline, e.g. from SA X can be sensed. The complement $\overline{\text{LBL}}$ of the SA Y is used only as a reference.

This sense amplifier architecture leads to different ways how information is stored in actual DRAM devices. In principle the following flavors exist here:

- **True-Cell** bit storage, where a logical `1` is always stored at bitline high voltage (e.g. 1.1 V) and a logical `0` is stored at low voltage (0 V).
- **Anti-Cell** bit storage, where a logical `1` is stored as bitline low voltage (0 V) and a logical `0` is stored at high voltage (1.1 V) value – i.e. the bit value is stored inverted.
- **Mixed-Cells** a combination of both, true and anti-cells depending on the address of the accessed cell.

This is, as mentioned before, due to different sense amplifier architectures, which use bitlines (LBL) and complement bitlines ($\overline{\text{LBL}}$) and different connection schemes to simplify the circuit structure in the DRAM arrays.

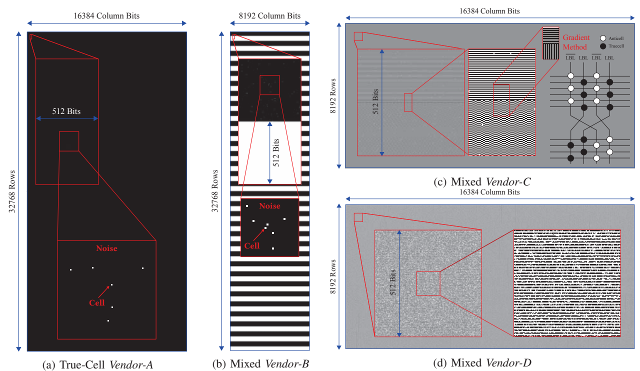


Figure 2: DDR3 DRAM Banks from Different Vendors for Reverse Engineering of True- and Anti-Cells

The vendor specific internal array architecture has a significant influence on the error behavior of DRAMs, and it plays, among other effects such as coupling a significant role for DPD. The storing flavor for a specific DRAM is usually unknown because it is the secret of each DRAM vendor. In Section IV we present a method, which can be used to determine the structure of true-, anti- and mixed-cells.

IV. Reverse Engineering of True- and Anti-Cells

The presented method works similar to the approach in [10]. First, a complete DRAM DIMM is filled with logical `1`s. Second, the DRAM's auto-refresh is disabled for 60,000-70,0000 seconds (16.7-19.4 hours), where for 3 hours the DRAM is additionally heated up to 80 °C in order to accelerate the charge loss and to remove the last residual charges i.e. noise, which distort our results. Third, the DRAM is read out and the data is visualized as a bitmap – black pixels represent a logical `0` and white pixels represent a logical `1`, respectively. If the DRAM consists of true-cells, the stored `1` has leaked and the bitmap is, beside some noise, fully black. If the DRAM uses anti-cells the logical `1` has been stored as a inverse (i.e. `0`), which cannot leak. Therefore, the bitmap is completely white. If the DRAM has mixed-cells, there will be white parts and black parts with some noise. Fig. 2 shows the results for DDR3 banks of the major DRAM vendors. Apparently, *Vendor-A* uses only true-cells and *Vendor-B*, *Vendor-C* and *Vendor-D* mix true- and anti-cells. For example, *Vendor-B* in Fig. 2b is alternating true- and anti-cells every 512 rows in a zebra-pattern. Furthermore, Fig. 2c shows that reverse engineering using the gradient method from [10] can reveal even more information: The figure shows that it is even possible to reconstruct detailed structures like a twist in the LBLs and $\overline{\text{LBL}}$s. These experiments show apparently that DRAMs have an asymmetric error behavior, which can be exploited for ECC.

V. Information Theoretic View

In this section we model DRAM retention errors by a noisy channel known from communications and use an information theoretic approach to evaluate the gain in using the Z-channel over using other channel models. In addition, we verify the advantage of the Z-channel by calculating the deviation of error occurrences compared with real DRAM retention error

measurements. The channel is the core of every communication system and models the characteristics of the transmission noise. A theoretical measure for the quality of a channel can be given by the *channel capacity* C, as introduced in Shannon's "A Mathematical Theory of Communication" [19]. C can be defined as the maximum rate of transmission, such that information can be transmitted reliably over the channel. An easy and frequently used discrete channel model is the *Binary Symmetric Channel* (BSC) in Fig. 3a. Without any knowledge of the internal retention error behavior of DRAMs, a BSC is assumed to estimate the errors. With the help of the binary entropy function

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p) \tag{1}$$

the channel capacity of the BSC can be written as

$$C_{BSC} = 1 - H(p), \tag{2}$$

where p is the crossover probability.

A different channel model is the *Binary Asymmetric Channel*, or *Z-channel*, that can be seen in Fig. 3b. In contrast to the BSC the probability for a $0 \rightarrow 1$ crossover is 0, resulting in the typical 'Z' structure that can be seen in the figure. Accordingly, the inverted Z-channel ($\bar{Z}$*-channel*) has a 0 probability for a $1 \rightarrow 0$ crossover. The Z-channel can be used to model a true-cell DRAM, the $\bar{Z}$-channel to model an anti-cell DRAM. The channel capacities of the Z-channel and the $\bar{Z}$-channel calculate to

$$C_Z = C_{\bar{Z}} = \log_2 \left(1 + (1-p) \cdot p^{\frac{p}{1-p}}\right). \tag{3}$$

For small probabilities p, Equation 3 can be rewritten as

$$C_Z = C_{\bar{Z}} \approx 1 - \frac{1}{2} H(p), \tag{4}$$

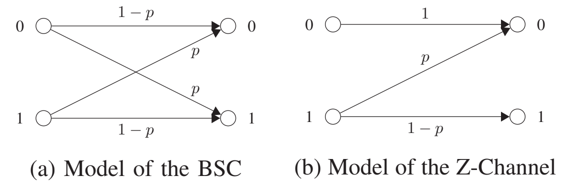
so obviously $C_Z > C_{BSC}$ [20].

A third possibility for a discrete channel is the *unidirectional channel*, sometimes denoted as *U-channel*. It behaves either like a Z-channel with crossovers only of type $1 \rightarrow 0$, or like a $\bar{Z}$-channel with crossovers only of type $0 \rightarrow 1$, and can be used to model a mixed-cell DRAM with a regular zebra-structure (like *Vendor-B* or *Vendor-D*). Its channel capacity is not yet fully determined, but can be bounded by

$$C_Z - \frac{1}{n} \leq C_Z - \frac{H(p)}{n} \leq C_U \leq C_Z, \tag{5}$$

where n is the length of the transmitted word, e.g. 64 for one DRAM burst element [21]. Thus, among the three presented channel models, the Z-channel has the highest channel capacity, which means, according to Shannon, that less redundancy is required to reliably transmit information over the Z-channel.

With the knowledge of DRAM's true- and anti-cell pattern derived in Section IV, every DRAM device can be regarded as a true-cell DRAM by inverting the bits that are stored in an anti-cell region. With this simple modification in the memory controller the retention error behavior of every DRAM can be modeled with the Z-channel. As shown before, this results in a large gain in channel capacity compared to assuming a U-channel (*Vendor-B*, *Vendor-D*), or a BSC, e.g. for *Vendor-C*, where true- and anti-cells are present in the same word



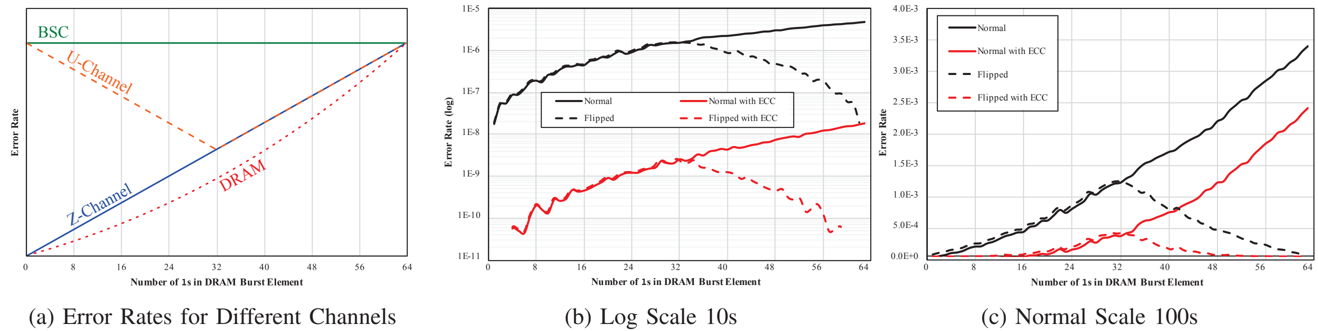
(a) Model of the BSC (b) Model of the Z-Channel

Figure 3: Channel Models

due to the LBL twist. In the remainder of this section, the deviation from the measured error distribution of the presented channel models is calculated to show that the Z-channel not only has the highest channel capacity, but also approximates the real DRAM retention error behavior best. The different channel models are depicted in Fig. 4a, where the x-axis represents the number of ones in a DRAM burst element, and the y-axis represents the error probability for some fixed retention time and temperature. The graph for the DRAM errors is derived from the measurements in Section VI. Although the DRAM errors are clearly not linear with respect to the x-axis, which is due to DPD and other overlapping cell leakage effects, the Z-channel approaches the measured errors best: By calculating the average error as a discrete integral of the shown curves in Fig. 4a, the Z-channel reduces the deviation from the measurements to 24.5%, whereas the U-channel assumes 87.7% and the BSC 149% more errors than actually occurred. Weighting the occurrences of the input with a binomial distribution, which corresponds to the natural distribution of the number of ones in a random bitstring, the Z-channel assumes 37.6% more errors compared to the measured error values, which is still less than the deviations of 49.2% for the U-channel and 171% for the BSC.

VI. A Z-Channel Aware DRAM Controller

In this section, a bit-flipping approach is presented, that can be used to largely increase DRAM's reliability with little overhead. The approach relies on the reverse-engineering procedure in Section IV and requires the DRAM controller to know the resulting pattern of true- and anti-cells, that is needed to assume a fully true-cell DRAM. As already pointed out, the Z-channel behavior of DRAM retention errors yields more errors happening when the corresponding burst element contains a large number of ones. Therefore, our approach flips the whole burst element whenever it contains more ones than zeros. The required one bit flip information per burst element, that is set to 1 if the burst is flipped, has to be secured along with the data. Therefore, the used Hamming code in ECC DRAM, that needs 8 bit redundancy to protect the 64 bit of data, is reduced to 7 bit redundancy to protect 65 bit of data. Obviously, this reduction of redundancy comes with a drawback: The extra parity bit is deleted, that is used to enable double error detection. Thus, upon occurrence of a two-bit error in the burst element, the reduced Hamming code will detect the error, but handle it as in the case of a correctable one-bit error, resulting in an additional error because of the decoder's attempt to correct a single-bit error. However, this behavior only affects the probability of undetected errors, but not the frame error rate. The results of the bit-flipping approach for a DDR4 true-cell DRAM of *Vendor-A* can be



(a) Error Rates for Different Channels (b) Log Scale 10s (c) Normal Scale 100s

Figure 4: DRAM Retention Errors

seen in Fig. 4b and Fig. 4c with refresh switched off for 10 s and 100 s, respectively. We apply randomized patterns to the entire DRAM device containing a specific number of `1`s in the DRAM write burst. First, we generate a random bit string of length equal to the device's burst size, with a fixed number of `1`s. Second, the DRAM is written completely with the previous determined bit string. Third, we disable the refresh, keep the DRAM at 25 °C and read out the data after a specific time. For each number of `1`s we repeat the above steps five times with shuffled data patterns. In our flip approach, the complete DRAM burst element is inverted if and only if it contains more than 32 `1`s, which results in a mirroring of the retention error curve. In addition to the results of the bit-flipping approach, also the error rates for ECC DRAM using SEC/DED, and the combination of both schemes are shown. With the bit-flipping approach, the error probability of DRAMs can be lowered significantly, increasing its reliability and allowing approaches like ADRAM. In some applications, where parts of the data contain a large number of ones, like applications using negative 2's complement numbers with a small dynamic range, the effect of the bit-flipping can even outperform the reliability increase through ECC (see Fig. 4c).

VII. Conclusion

In this paper a channel model for DRAM retention errors was presented. It takes DRAM's inherent asymmetric error behavior into account, that relies on the fact that charge can only leak out of the cell, but not in again. With the results of our reverse-engineering procedure, every DRAM can be regarded as a true-cell DRAM, allowing the usage of custom Z-channel ECC codes to protect the data and increase DRAM's reliability. A simple bit-flipping approach can be used to exploit the asymmetry to improve the reliability with little overhead.

Acknowledgment

The authors thank Martin Zeyen for his support. This work was initiated in the context of a cooperation with Huawei. The project OPRECOMP (http://oprecomp.eu) acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the European Union's Horizon 2020 research and innovation programme, under grant agreement No 732631. This work was also supported by the Carl-Zeiss Stiftung.

References

[1] M. Jung, et al. *Efficient Reliability Management in SoCs - An Approximate DRAM Perspective*. In 21st Asia and South Pacific Design Automation Conference (ASP-DAC), 2016.

[2] S. Liu, et al. *Flikker: Saving DRAM Refresh-power Through Critical Data Partitioning*. SIGPLAN Not., 46(3):213–224, March 2011.

[3] J. Lucas, et al. *Sparkk: Quality-Scalable Approximate Storage in DRAM*. In The Memory Forum, June 2014.

[4] A. Raha, et al. *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.

[5] A. Raha, et al. *Quality Configurable Approximate DRAM*. IEEE Transactions on Computers, 66(7):1172–1187, July 2017.

[6] G. Stazi, et al. *Introducing approximate memory support in Linux Kernel*. In 2017 13th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), pages 97–100, June 2017.

[7] J. A. Halderman, et al. *Lest We Remember: Cold-boot Attacks on Encryption Keys*. Commun. ACM, 52(5):91–98, May 2009.

[8] K. G. Paterson, et al. *A Coding-Theoretic Approach to Recovering Noisy RSA Keys*, pages 386–403. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[9] M. Hassan, et al. *Reverse-engineering embedded memory controllers through latency-based analysis*. In 21st IEEE Real-Time and Embedded Technology and Applications Symposium, pages 297–306, April 2015.

[10] M. Jung, et al. *Reverse Engineering of DRAMs: Row Hammer with Crosshair*. In International Symposium on Memory Systems (MEMSYS 2016), 2016.

[11] P. Pessl, et al. *DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks*. In 25th USENIX Security Symposium (USENIX Security 16), pages 565–581, Austin, TX, August 2016. USENIX Association.

[12] R. W. Hamming. *Error Detecting and Error Correcting Codes*. Bell System Technical Journal, 26(2):147–160, 1950.

[13] H. J. Kwon, et al. *23.4 An extremely low-standby-power 3.733Gb/s/pin 2Gb LPDDR4 SDRAM for wearable devices*. In 2017 IEEE International Solid-State Circuits Conference (ISSCC), pages 394–395, Feb 2017.

[14] C. K. Lee, et al. *23.2 A 5Gb/s/pin 8Gb LPDDR4X SDRAM with power-isolated LVSTL and split-die architecture with 2-die ZQ calibration scheme*. In 2017 IEEE International Solid-State Circuits Conference (ISSCC), pages 390–391, Feb 2017.

[15] J. M. Berger. *A note on error detection codes for asymmetric channels*. Information and Control, 4(1):68–73, 1961.

[16] M. Neagu, et al. *Unidirectional error detection, localization and correction for DRAMs: Application to on-line DRAM repair strategies*. In IEEE 17th International On-Line Testing Symposium (IOLTS), pages 264–269. IEEE, 2011.

[17] B. Narasimham et al. *A multi-bit error detection scheme for DRAM using partial sums with parallel counters*. In IEEE International Reliability Physics Symposium, IRPS 2008, pages 202–205. IEEE, 2008.

[18] K. Kim et al. *A New Investigation of Data Retention Time in Truly Nanoscaled DRAMs*. Electron Device Letters, IEEE, 30(8):846–848, Aug 2009.

[19] C. E. Shannon. *A Mathematical Theory of Communication*. Bell System Technical Journal, 27:379–423 and 623–656, July–October 1948.

[20] L. G. Tallini, et al. *On the capacity and codes for the Z-channel*. In IEEE International Symposium on Information Theory, page 422. IEEE, 2002.

[21] L. G. Tallini. *Bounds on the capacity of the unidirectional channels*. IEEE Transactions on Computers, 54(2):232–235, 2005.