

Efficient Coding Scheme for DDR4 Memory Subsystems

Kira Kraft

Technische Universität Kaiserslautern
Kaiserslautern, Germany
kraft@eit.uni-kl.de

Deepak M. Mathew

Technische Universität Kaiserslautern
Kaiserslautern, Germany
deepak@eit.uni-kl.de

Chirag Sudarshan

Technische Universität Kaiserslautern
Kaiserslautern, Germany
sudarshan@eit.uni-kl.de

Matthias Jung

Fraunhofer IESE
Kaiserslautern, Germany
matthias.jung@iese.fraunhofer.de

Christian Weis

Technische Universität Kaiserslautern
Kaiserslautern, Germany
weis@eit.uni-kl.de

Norbert Wehn

Technische Universität Kaiserslautern
Kaiserslautern, Germany
wehn@eit.uni-kl.de

Florian Longnos

Huawei Technologies Co. Ltd,
Data Center Technologies Lab
Hangzhou, China
florian.longnos@huawei.com

ABSTRACT

DRAMs face several major challenges: On the one hand, DRAM bit cells are leaky and must be refreshed periodically to ensure data integrity. Therefore, DRAM devices suffer from a large overhead due to refreshes both in terms of performance (available bandwidth) and power. On the other hand, reliability issues caused by technology shrinking are becoming a large concern. Thus, ECC techniques for DRAM errors, and especially for retention errors, gain more and more importance. In this paper, we present an investigation on DRAM errors and derive a detailed model for these types of errors. The model is verified by various measurements, and analyzed from an information theory point of view. Based on this model, a scheme is presented that largely improves DRAM's reliability with low overhead.

CCS CONCEPTS

• **Hardware** → **Dynamic memory; Error detection and error correction;**

KEYWORDS

DRAM, Error Correcting Codes (ECC), Error Mitigation, DBI

ACM Reference Format:

Kira Kraft, Deepak M. Mathew, Chirag Sudarshan, Matthias Jung, Christian Weis, Norbert Wehn, and Florian Longnos. 2018. Efficient Coding Scheme for DDR4 Memory Subsystems. In *The International Symposium on Memory Systems (MEMSYS)*, October 1–4, 2018, Old Town Alexandria, VA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3240302.3240424>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MEMSYS, October 1–4, 2018, Old Town Alexandria, VA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6475-1/18/10...\$15.00

<https://doi.org/10.1145/3240302.3240424>

1 INTRODUCTION

Dynamic Random Access Memories (DRAMs) are key components in any computing system. Due to the large increase in data volume in many applications, such as machine learning, image processing, data analytics etc., DRAMs largely impact the overall system performance and power consumption. E.g. it was shown that in natural language processing more than 80% of the overall power consumption in state-of-the-art computing systems originated from DRAMs. Moreover, in many applications, especially machine learning, the overall system performance is often limited by the memory bandwidth and not by the computation.

Due to shrinking technology nodes, reliability is a major concern in DRAMs. Thus, many techniques exist to improve yield and reliability. Examples are circuit redundancy that is activated by laser or electrical fuses after fabrication or the use of *Error Correcting Codes* (ECC). In this paper, we focus on retention errors, that is, transient failures due to the leakage of the stored charge on the capacitance in the memory cell. To mitigate these errors, regular refresh of the storage cell has to be performed. It is shown that in future DRAM devices the power consumption of the refresh operation will account up to 50% of the total DRAM power [4, 22]. Additionally, the effective memory bandwidth is reduced by the refresh command. Thus, the refresh process exacerbates the power and bandwidth challenge of DRAMs. One possibility to reduce this overhead in power and bandwidth is lowering the refresh rate. However, this may increase the likelihood of retention errors. ECC techniques can help to correct these errors but the overhead of ECC must be small to not wipe out the gain achieved by the reduction of the refresh rate. Each storage cell experiences a mixture of different leakage currents. In addition, process defects can reduce the maximum amount of charge that can be stored. So, the retention behavior of a single memory cell depends on many parameters, such as process variations, temperature, activation energy, previous charge level, cell structure, and data pattern dependency [1].

From an information theory point of view, a memory cell can be considered as a noisy channel. Traditional ECC techniques for DRAMs assume a symmetric behavior of the retention errors. In

this paper, we present a more accurate error model for the retention behavior with a special focus on the dependency on the cell structure (true- or anti-cell). We show that the channel capacity (according to Shannon’s definition for noisy channels) of a single memory cell is larger than in the traditional commonly used symmetrical model. Using this insight, we derive a low-overhead coding scheme that improves the retention behavior. This coding scheme can be easily combined with the existing *Data Bus Inversion* (DBI) mechanism of state-of-the-art DDR4 memories and the new scheme does not require any changes in the DRAM itself. We verify our new approach with a large set of experiments testing various parameters, and give a characterization of applications that can benefit largely from the proposed scheme.

In detail, this paper makes the following contributions:

- We present a detailed model for DRAM retention errors, including different kinds of crosstalk and leakage, and the combination of both. Various measurements will verify this model and the asymmetry of DRAM retention errors.
- The error model is analyzed from an information theoretic point of view. We show that the channel capacity is higher than commonly assumed, which allows for better and tailored coding schemes.
- We propose one such coding scheme to make the storage of data in the DRAM more reliable. The presented coding scheme can be combined with the existing DBI protocol in DDR4 memories, which makes it very low-overhead. For different applications, the proposed coding scheme can largely outperform the commonly used ECC scheme, thereby using the same amount of parity information.

The rest of the paper is organized as follows: In Section 2, we review the related work. Section 3 gives the necessary background on DRAMs and the DDR4 DBI mechanism. A special focus of this section is the explanation of leakage paths and crosstalks happening in the DRAM. In Section 4, an advanced model for DRAM errors built on the aforementioned error sources is presented and confirmed with detailed measurements. Section 5 puts the derived error model into an information theory channel model and shows the benefit of using this new channel model for DRAM retention errors. Our proposed low-overhead coding scheme is described in Section 6. Section 7 describes the experimental setup, while the results are presented and discussed in Section 8. The paper is finally concluded in Section 9.

2 RELATED WORK

This section reviews the related work on DRAM error models and current ECC schemes.

2.1 DRAM Retention Profiling and Error Models

One of the first studies on DRAM retention time even in combination with ECC, before the DDR era, was presented by Katayama et al [15]. In 2009, Kim and Lee presented a detailed study on data retention times of nanoscaled DDR DRAMs [16]. Most of the following investigations, for instance [22, 24, 27] and [20], are based on these results. However, Jung et al. measured several DDR3 DRAM devices from 2009 of the same vendor [12, 13] and confirmed that

these presented numbers in [16] are way too pessimistic in the average case. Liu et al. [21] performed a detailed analysis of data retention behavior of DDR3 DRAMs with respect to temperature, *Data Pattern Dependency* (DPD), and *Variable Retention Time* (VRT). However, their analysis is limited to short refresh intervals (< 10 s). The authors of [7] also measure only for short refresh intervals (< 3 s) and only one bank of one single DDR3 DRAM. Furthermore, retention data on LPDDR2, LPDDR3 and LPDDR4 has been published in [2], [37] and [29] respectively, but for short refresh intervals. In 2015, [38] conducted measurements on Wide I/O DRAMs and DDR3 DRAMs, and presented a retention error model. But their measurements are also for very short refresh intervals (up to 2 s). Most recently, [25] performed retention analysis on state-of-the-art DDR4 DRAMs, but the focus of their study was to analyze the influence of true-/anti-cell architecture on DRAM retention behavior. In contrast to the previous works, our retention analysis considers the influence of crosstalks (DPD) and cell leakage over a wide range of refresh intervals (0.1 s to 1000 s) and temperatures (30°C, 60°C, 90°C).

2.2 ECC in DRAM

A state-of-the-art ECC DRAM DIMM, for instance, consists of 8 DRAM devices and a 9th device, that is 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* (SEC/DED) code [6]. Oftentimes, a variation of the Hamming code is used, the so called Hsiao code [8], that aims at minimizing the implementation complexity of the SEC/DED code. Another well-known approach is IBM’s Chipkill [5], that can restore a full chip failure. Two vendors recently introduced on-die ECC [18, 19] to avoid retention errors. They lower the refresh rates by a factor of $4\times$ in order to reduce the power consumption. This shows the importance of ECC for retention errors in DRAMs.

In contrast to the above mentioned ECC schemes that do not presume any asymmetry in the DRAM retention errors, the authors of [17] introduced a coding scheme that relies on this asymmetry. However, their underlying error model only focuses on DRAM cell leakage, but not on crosstalk effects and DPD. In this work, we will present a more sophisticated error model (see Section 4) that is validated by a large set of measurements.

3 DRAM BACKGROUND

In the following we present the DRAM device architecture and functionality to understand how information is stored in the DRAM cells. Afterwards, the DBI mechanism of state of the art DDR4 DRAMs will be reviewed. Finally, a detailed explanation of different leakage paths and crosstalk effects that can cause retention errors is given.

3.1 Basic Architecture

As depicted in Figure 1a, a DRAM device is organized as a set of memory banks (e.g. eight) that include independent memory arrays (e.g. two). Each memory array has row and column decoders, master wordline drivers and *Secondary Sense Amplifiers* (SSA). Buses, buffers, control signals, voltage regulators, charge pumps and other peripherals are shared between the different banks. The memory

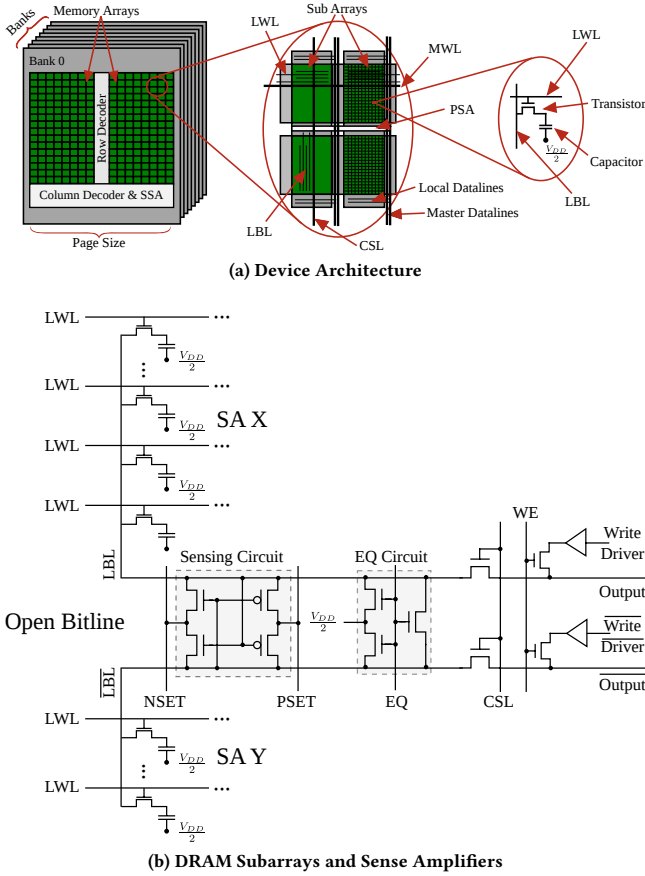


Figure 1: DRAM Architecture

arrays are further split in a hierarchical structure out of *Subarrays* (SA) for efficient wiring, increased speed and reduced power consumption. Therefore, each SA is equipped with *Primary Sense Amplifiers* (PSA).

A single memory cell is built as a transistor capacitor pair where the data is stored in the capacitor as a charge. A DRAM cell has various leakage paths, discussed in Subsection 3.3, which degrade the stored charge over time. Therefore, each cell in the DRAM must be refreshed every t_{REF} ($t_{REF} = 64$ ms for JEDEC compliant DRAMs) in order to prevent so called *Retention Errors*. The individual cells in each SA are connected to *Local Wordlines* (LWL) and *Local Bitlines* (LBL). The LBLs and LWLs are connected to global *Master Bitlines* (MBL) and *Master Wordlines* (MWL), respectively, which span over the complete memory array.

The SAs are connected to a sensing circuit (PSA) as depicted in Figure 1b. This shows an open bitline configuration in which a PSA is connected to two Local Bitlines (LBL) coming from SA X and SA Y. It is important to know that only one bitline, e.g. from SA X can be sensed at a time, while the other one (the complement $\overline{LBL}$ of 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 at 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 ($\overline{LBL}$) and different connection schemes to simplify the circuit structure in the DRAM arrays.

The vendor specific internal architecture of true- and anti-cells has a significant influence on the error behavior of DRAMs [12, 25], and it plays, among other effects such as coupling, a significant role for *Data Pattern Dependency* (DPD). The storing flavor for a specific DRAM is usually unknown because it is a proprietary of each DRAM vendor. However, it is possible to reverse engineer the DRAMs to determine the structure of true-, anti- and mixed cells [14, 17].

3.2 DBI in DDR4 DRAMs

DDR4 DRAMs support *Data Bus Inversion* (DBI) to save power and to reduce the simultaneous switching noise [11, 35]. It is shown that a typical memory usage of 70% Reads to 30% Writes yields about 27% system power savings when DBI is enabled [36]. This is an optional feature which can be enabled in the DRAM's *Mode Register Settings* (MRS). DDR4 supports DBI in write direction, read direction and both write and read direction, with one *dbi_n* pin for every byte lane. Once DBI is enabled in the MRS it works as follows:

- For reads, the DRAM checks if more than four bits of a byte lane are low. If yes, then it will invert the data on that byte lane and drive the *dbi_n* pin low. If four or less number of bits in a byte lane are low it sends the original stored data and drives the *dbi_n* pin high. A value of *dbi_n* = 0 indicates that the data was sent inverted, and needs to be inverted again at the controller PHY.
- For writes, the DRAM controller PHY checks the number of bits which are low. If more than four bits in a byte lane are low, PHY will invert the byte lane and drive the *dbi_n* pin low, so the DRAM has to invert the data back internally before storage. If four or less number of bits are low, no data is inverted and *dbi_n* remains high.

3.3 Cell Leakage in DRAM

In this subsection, we describe the various leakage paths in a DRAM cell [23, 30], and their influence on the stored charge: a stored 0 or 1. As shown in Figure 2, the following leakage paths exist in a DRAM cell:

- **Drain Leakage (1)**, which includes the P-N junction leakage as well as *Gate Induced Drain Leakage* (GIDL). GIDL is

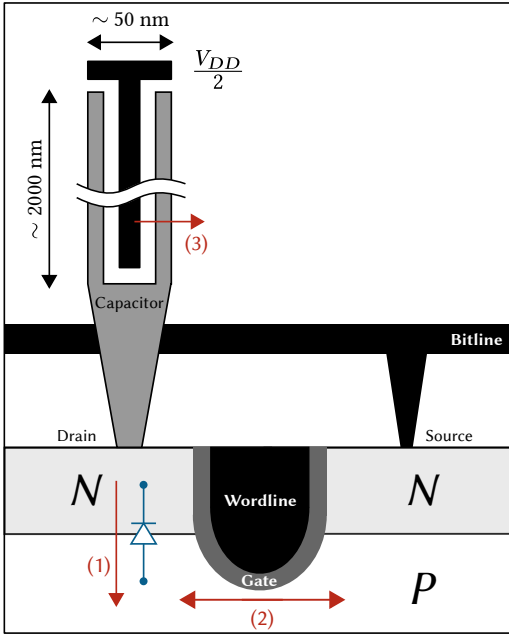


Figure 2: Leakage Paths in Modern Buried Wordline DRAM Architecture [31]

mainly caused by *Trap Assisted Tunneling* (TAT), and it is influenced by the number and distribution of traps in the band-gap region as well as the electric field. Since the negative wordline voltage and the positive charge stored in the cell capacitor (when a 1 is stored in the cell) increase the electric field in the band-gap region (gate-drain overlap region), GIDL is the major source of leakage for a stored 1 in the DRAM cell [26].

- **Sub-threshold Leakage (2)**, which is the drain-source leakage of the cell transistor when it is in the OFF state. This current depends on various factors such as negative wordline voltage, bulk voltage, etc. When the bitlines are in precharged state ($V_{CC}/2$) this can slightly charge the cell capacitor and therefore cause the degradation of a 0 stored in the cell. It can also degrade a 1 stored in the cell by discharging to the bitline, but the leakage will be very small due to the increased Threshold Voltage (V_t) of the access transistor when a 1 is stored (Body-Bias effect).
- **Cell Capacitor Leakage (3)**, which is the leakage through the cell capacitor dielectric. With the technology scaling, also the capacitor area is decreasing. Therefore, to maintain the cell capacitance at the previous value, dielectric thickness has to be reduced, which increases the leakage. The use of new *Metal Insulator Metal* (MIM) structure with high-k dielectric materials [9] has helped to reduce this leakage. Capacitor leakage influences both stored 0's and stored 1's.

3.4 Crosstalks in DRAM

Crosstalks can either disturb the data stored in the cells or disturb their sensing. Following are the major sources of crosstalks in DRAM [21, 28, 32, 39, 40]:

- **Bitline to Bitline Coupling** disturbs the pre-sensed charge on a bitline, thereby affecting the sense margin. This is the major source of crosstalk in modern DRAM architectures because all bitlines are running in parallel and are switched together. This form of crosstalk creates sensing errors, especially when the charge in a cell is degraded due to various leakage paths described in 3.3.
- **Wordline to Wordline Coupling** causes disturbance on neighboring wordlines when a wordline is activated. This will increase the sub-threshold leakage of the cells connected to the neighboring affected wordlines and therefore degrade the value stored in the cell.
- **Wordline to Bitline Coupling** can also degrade the value stored in the DRAM cells, but the amount of this coupling capacitance is very small since the bitlines are perpendicular to wordlines. Moreover, this form of coupling is drastically reduced in modern buried wordline architectures [31].
- **Bitline to Cell Coupling** exists between bitlines and storage node contact [10]. The use of low-k filling or air-gap filling helps to reduce this coupling.

4 DRAM ERROR MODEL

In this section we present an advanced error model for the retention behavior of DRAMs. Our modeling approach considers the influence of cell leakage, crosstalks, the stored data value (0 or 1) and the stored data in the neighboring cells (*Data Pattern Dependency*) on data retention errors.

First, we analyze the data pattern dependency of retention errors over a wide range of refresh intervals (100 ms to 1000 s) and at different temperatures. DRAM retention errors originate from cell leakage as well as crosstalks. The cell leakage degrades the charge stored in a cell, while crosstalks flip the degraded cell charge on the bitline during the sensing process. For short refresh intervals (<64 ms), the cell has a strong charge (either a full charge or no charge), and crosstalks have less influence on the sensing process. But when the charge in a cell is weak (either the 1 is degraded to $1-\epsilon$ or the 0 is degraded to $0+\epsilon$), crosstalks can flip the value on the bitline, and therefore cause retention failures. Since the crosstalks are highly *Data Pattern Dependent* (DPD), retention failures in this region also show similar data pattern dependencies. When the charge in a cell is degraded very much (longer refresh intervals), crosstalks have much less influence in the sensing process since the charge in many cells has already crossed the required level for correct sensing (sensing threshold). Therefore there is hardly any DPD in this region.

Figure 3 plots the retention behavior of a true-cell DDR4 DRAM for two different data patterns: 0xFF and random. The measurements were conducted with the measurement platform described in Section 7. At room temperature (30°C) and for lower refresh intervals (<50 s) the random pattern shows more number of bitflips than 0xFF pattern. The larger number of bitflips in this region for random data pattern (high DPD) is caused by various crosstalks described in

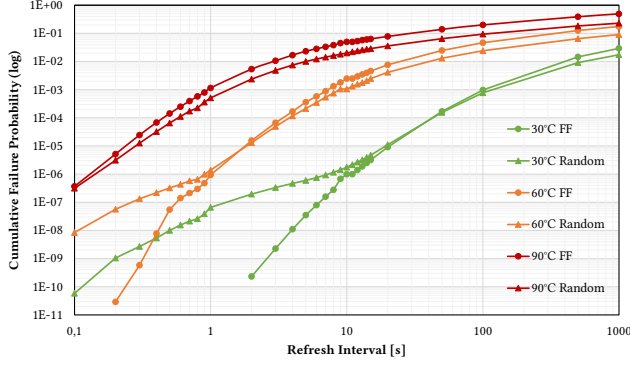


Figure 3: Data Pattern Dependency of Retention Errors

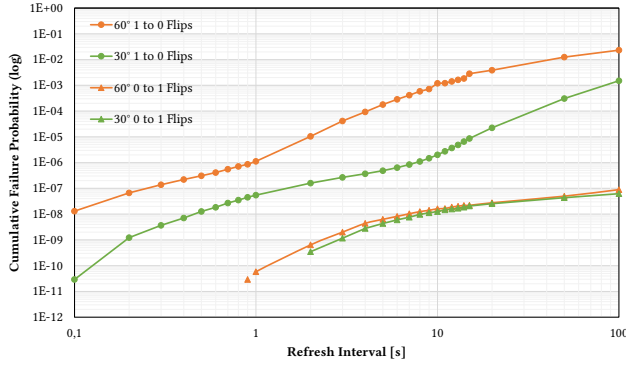


Figure 4: Failure Probabilities for 0's and 1's

Section 3.4. There is a crossover happening in the retention curves of 0xFF and random data pattern at 50 s. Afterwards, 0xFF pattern gives more retention errors than the all random pattern. This is because of the low DPD in this region since many of the cells have already crossed the sensing threshold. As temperature increases, cell leakage goes high. Therefore, the crossover point between the DPD region and the non-DPD region appears earlier for higher temperatures.

Second, we model the failure probabilities of a stored 0 and a stored 1 in a true-cell DRAM. As explained in Section 3.3 and Section 3.4, a 1 can flip to 0 mainly due to drain leakage and crosstalks. Similarly, a 0 can flip to 1 due to sub-threshold leakage and crosstalks. However, since the sub-threshold leakage is very small compared to the drain leakage, the probability of a 0 to 1 failure is very small. To prove this experimentally, we conducted measurements on off-the-shelf DDR4 DRAMs (true-cell devices) at different temperatures. Figure 4 plots the probability of 1 to 0 and 0 to 1 crossovers for a random data pattern. It shows that the probability of a 0 to 1 flip in the DRAM is very small, and doesn't vary much with temperature. E.g., among all errors, only 0.005% contribute to 0 to 1 errors (60°C, 1 s refresh interval), so the DRAM retention errors behave highly asymmetrical.

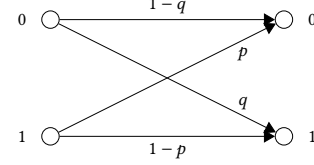


Figure 5: Model of the Binary Channel

5 INFORMATION THEORETIC MODEL

In this section we model DRAM errors by a noisy channel known from communications and evaluate its benefit with an information theoretic approach. The channel model takes the error behavior identified in the previous section into account.

The channel is the core of every communication system and models the characteristics of the transmission noise. In our case, the storage of the data inside the DRAM will be considered as a noisy channel. 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" [33], that laid the foundation of information theory. C can be defined as the maximum rate of transmission, such that information can be transmitted reliably over the channel. A very general discrete channel model is the *Binary Channel* (BC) in Figure 5 with crossover probabilities q and p . With the help of the binary entropy function

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x) \quad (1)$$

for $0 \leq x \leq 1$, the channel capacity of the binary channel can be written as

$$C_{BC} = \frac{q}{1-q-p} \cdot H(p) - \frac{1-p}{1-q-p} \cdot H(q) + \log_2 \left(1 + 2^{\frac{H(q)-H(p)}{1-q-p}} \right), \quad (2)$$

with $q \leq p$.

We will now consider two special cases of the binary channel, namely the *Binary Symmetric Channel* (BSC) and the *Z-Channel*, which is a fully asymmetric channel. The BSC is defined via equal crossover probabilities p and q , so $p = q$. From Equation 2 its channel capacity can easily be calculated to

$$C_{BSC} = 1 - H(p). \quad (3)$$

In contrast, the Z-channel has the property, that $q = 0$, so no 0 to 1 crossover is possible. Removing this arc from Figure 5 results in the typical 'Z' structure. Its channel capacity can be calculated with $q = 0$ from Equation 2 to

$$C_Z = \log_2 \left(1 + (1-p) \cdot p^{\frac{p}{1-p}} \right), \quad (4)$$

which can, according to [34], be approximated for small p to

$$C_Z \approx 1 - \frac{1}{2} H(p). \quad (5)$$

Thus, the channel capacity of the Z-channel is larger than the capacity of the BSC. According to Shannon [33], this means that less redundancy is required to reliably transmit information over the Z-channel, compared to the BSC.

As discussed in Section 4, DRAM errors have a highly asymmetric behavior (with a known cell structure), meaning that almost all errors occurring are 1 to 0 errors for a true-cell DRAM. As can be seen in Figure 4, only a negligible amount of errors contribute to 0 to 1 errors, such that $q \approx 0$ in the BC model. Therefore, we can assume a Z-channel to model these errors. With the data plotted in Figure 4, we can observe crossover probabilities $p = 9 \cdot 10^{-6}$ and $q = 4.6 \cdot 10^{-10}$ for a single cell in a random data pattern for 1 s refresh interval at 60°C. Under these probabilities, the channel capacity calculated from Equation 2 equals the capacity calculated from Equations 4 and 5,

$$C_{DRAM-BC} = 0.999918 = C_{DRAM-Z}.$$

If the DRAM consists of both true- and anti-cells, the reverse-engineering technique from [17] can be used to identify the corresponding regions and to logically reduce to the case of a true-cell DRAM.

As already pointed out, the higher capacity of the Z-channel motivates the search for better coding schemes, that have to exist according to Shannon. One such coding scheme is the Berger code [3], that can identify *all* asymmetric errors. In systems, where the DRAM is used as a cache, and erroneous data in the DRAM can be loaded again from another memory, the Berger code is a very powerful tool to fully protect data from asymmetric errors. In Section 6, we will present a new low overhead coding scheme, that relies on the DRAM's Z-channel property and can largely increase DRAM's reliability.

6 PROPOSED METHOD

In this section, the proposed coding scheme based on the in Section 4 derived DRAM error model will be presented. Since in the Z-channel, only a 1 can be affected by errors (for true-cells), the idea is to store data inside the DRAM zero-dominant. Accordingly, the data is stored one-dominant for anti-cells. Therefore, the existing DBI mechanism is used to possibly invert the data at the DRAM (byte-wise), depending on the number of ones inside each byte of the DRAM burst element. As already pointed out, the structure of true- and anti-cells can be reverse-engineered using the procedure presented in [17].

To this end, the DBI mechanism of DDR4 DRAMs is combined with the vendor specific internal structure of true- and anti-cells. Since DBI assures that each 8 bit of data are transferred one-dominant via the bus, to exploit the Z-channel property of the DRAM, the data has to be inverted again when stored in a true-cell region, where zero-dominant data is less error-prone. Therefore, the existing dbi_n signal is used to tell the DRAM whether or not to flip the data coming from the bus, resulting in $dbi_n = 0$ for true-cell regions, and $dbi_n = 1$ for anti-cell regions.

To retrieve the correct data when transferring it back to the controller, an extra ECC bit needs to be stored, that indicates whether the data stored in the DRAM was the original data or not. Whenever data is read from the DRAM, the according ECC bit is loaded and XORed with the current dbi_n to get the information whether the data has to be inverted or not. Note that this requires 8 ECC bits per DRAM burst element (64 bit), so the overhead is the same as for the traditional Hamming SEC/DED code.

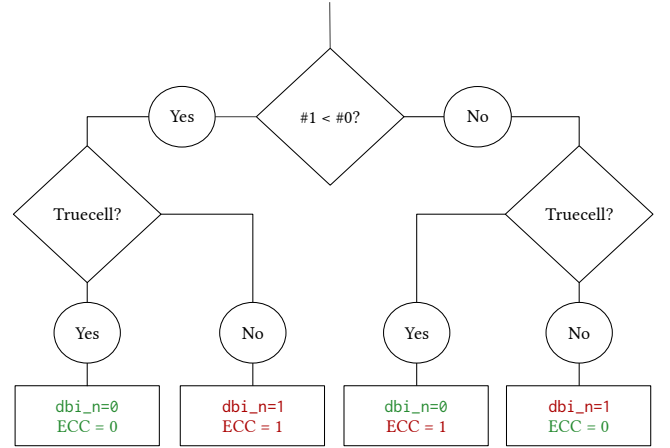


Figure 6: Flowchart of the Proposed Method

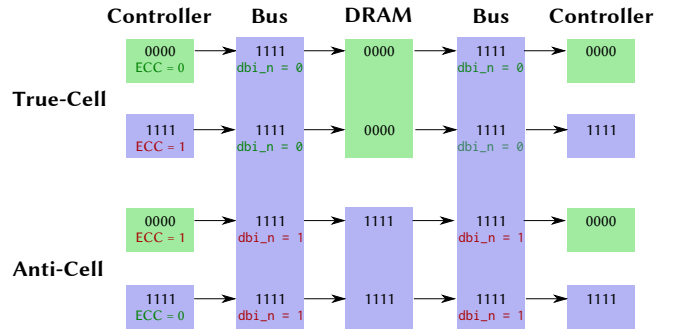


Figure 7: Example of the Proposed Method

The flowchart in Figure 6 can be used to set the ECC bit and the dbi_n signal for each 8 bit of data.

In addition, different example cases for the scheme are depicted in Figure 7 with all-zero and all-one data for true- and anti-cell regions. For simplification, only four bit of data are shown here, whereas the DBI works byte-wise. Consider, e.g., the case of the all-one pattern for the true-cell region. Since the true-cell DRAM has a higher reliability with storing the all-zero pattern, the data will be stored inverted, resulting in ECC = 1. In addition, since the data is transferred one-dominant via the bus, it has to be inverted at the DRAM, resulting in $dbi_n = 0$ in write direction. In read direction, the data is flipped at the DRAM to be transferred as the all-one pattern via the bus. Accordingly, dbi_n is set to 0 in read direction. Since $ECC \oplus dbi_n = 1$, the data does not have to be inverted again at the controller.

If DBI is only enabled in write direction, but not in read direction, the data can still be retrieved correctly on the read path by setting $dbi_n = 0$ for true-cell regions, and $dbi_n = 1$ for anti-cell regions. Accordingly, for DBI enabled in write and read direction, errors can be detected in cases where $dbi_n = 0$ ($dbi_n = 1$) in read direction, but the corresponding address belonged to an anti-cell (true-cell) region, respectively.

In addition, the error mitigation of this coding can be combined with ECC, like the SEC/DED Hamming code to further improve

DRAM’s reliability. In the following section, we will explain the experimental setup, followed by the experimental results on the benefit of the proposed method for different applications. Note that the proposed scheme is no ECC scheme in the classical way since neither error correction, nor error detection are performed. In contrast, it can effectively mitigate transient failures due to leakage and crosstalk effects. As shown in Section 8, the scheme can also be combined with a classical ECC to be able to detect and correct all kinds of errors occurring.

7 EXPERIMENTAL SETUP

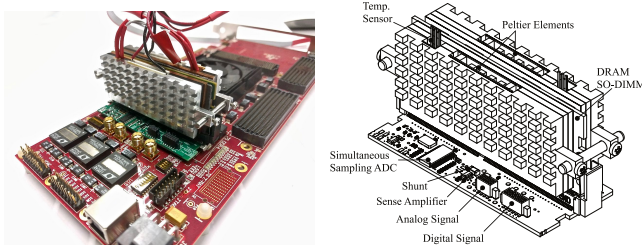


Figure 8: Measurement Platform and Adapter Board

Figure 8 shows the experimental platform for performing retention measurements. It is designed to measure retention errors, power consumption, and to heat up the DRAM devices of DDR4 SO-DIMM modules. The heating section consists of a mechanical setup, which is placed on the surface of the DRAM devices to heat them up within a range of 25°C to 95°C. The accuracy of the temperature control was determined to $\pm 2^\circ\text{C}$ using thermal simulations.

The adapter board is designed to have the same dimensions as a regular DDR4 SO-DIMM. This way it can be directly plugged in to almost any kind of DDR4 host system. For our tests we used a Xilinx Virtex Ultrascale FPGA-based evaluation platform. The standard *Memory Interface Generator* (MIG) memory controller from Xilinx is customized to generate the required command and data sequences for the measurements and to disable the refreshes. A *Virtual Input/Output* (VIO) core connected to the custom MIG enables the real-time monitoring and control of the internal FPGA signals. Furthermore, we add the functionality to measure the error rates of single devices on an SO-DIMM by separating the data signals, which is important for the following measurements. The entire measurement setup is fully automated using TCL scripts running in the Vivado Hardware Manager.

The random data pattern was generated in the memory controller using *Linear Feedback Shift Registers* (LFSRs). There are 8 LFSRs with different seeds. Each LFSR generates a new random data for every *WRITE* command to the DRAM. VIO was used to set the refresh interval, and to control various signals such as enable/disable ECC, enable/disable the proposed method, etc. Each test was performed by first writing the complete DRAM with a specific data pattern, then waiting for a certain duration with refresh switched-off, and finally, reading back the written data from the entire DRAM. The read data is compared against the written data to count the number of retention failures.

8 EXPERIMENTAL RESULTS

In this section, we conducted measurements on our platform described in Section 7 for 4 different test benches. Each 64 bit of data contain a varying part of random data (8 to 64 bit), while the rest is padded with either 1’s or 0’s to mimic an application using 2’s complement numbers with a small dynamic range. An application using such kind of data can be found in signal processing.

The results for the different data are depicted in Figures 9-12, where Figure 9 refers to an application using a dynamic range of 8 bit, Figure 10 and Figure 11 show the results for 16 and 32 bit dynamic range, respectively, and Figure 12 depicts the measurements with random data. The x-axis shows the time that the data stayed in the DRAM without refresh, the y-axis the probability of a bit-error. In all figures, the green curve corresponds to uncoded data, the orange curve shows the error probability when securing the data with a SEC/DED Hamming code, and the red curve depicts the results of our proposed method from Section 6. The gray curve was obtained by combining a modified inversion approach with a modified ECC scheme as follows: First, the SEC/DED Hamming code is reduced by its extra parity bit which enables the double error detection. Note that this largely increases the probability of an undetected error, but only slightly the bit-error rate: In case of a double error occurring, the code will now make an attempt to correct it, which will result in a third error introduced. Removing the double-error detection of the Hamming code should therefore only be performed in non-critical applications, that do not rely on the error detection. Second, the bit that was used as extra parity bit before now stores the information of the proposed method with a flip granularity of 64 bit (instead of 8 bit). To this end, the DBI mechanism must be modified heavily to make a decision whether or not to flip the data on an 64 bit granularity. As a result, each 64 bit of data are stored inverted or not, depending on its number of 1’s and the region in the DRAM in which the data is stored.

As can be observed in the figures, the proposed method outperforms ECC especially for the applications with small dynamic ranges (Figure 9 and 10), where up to 2 orders of magnitude can be obtained compared to the SEC/DED ECC (e.g. Figure 9b). In addition, the modified approach depicted in the gray curves shows significant improvement over both the proposed method and the traditional SEC/DED ECC. Especially the improvement over the Hamming SEC/DED code in Figures 9 - 11 shows that the proposed scheme can exploit the higher capacity of the Z-channel. For the random data in Figure 12, nearly no improvement was possible with our proposed method. This behavior is due to the uniform distribution on the number of 1’s and 0’s inside the data. Therefore, an inversion of random data should still almost behave like random data.

Overall, the proposed method and its modification combined with the reduced Hamming code can largely reduce the error probability in recent DDR4 DRAMs, and therefore significantly increase their reliability. Note, again, that the proposed scheme is not a error correcting code in the classical way, but focuses on making the data robust to retention errors due to leakage and crosstalks. In addition, it can be combined with a classical ECC to be able to detect and correct all kinds of errors.

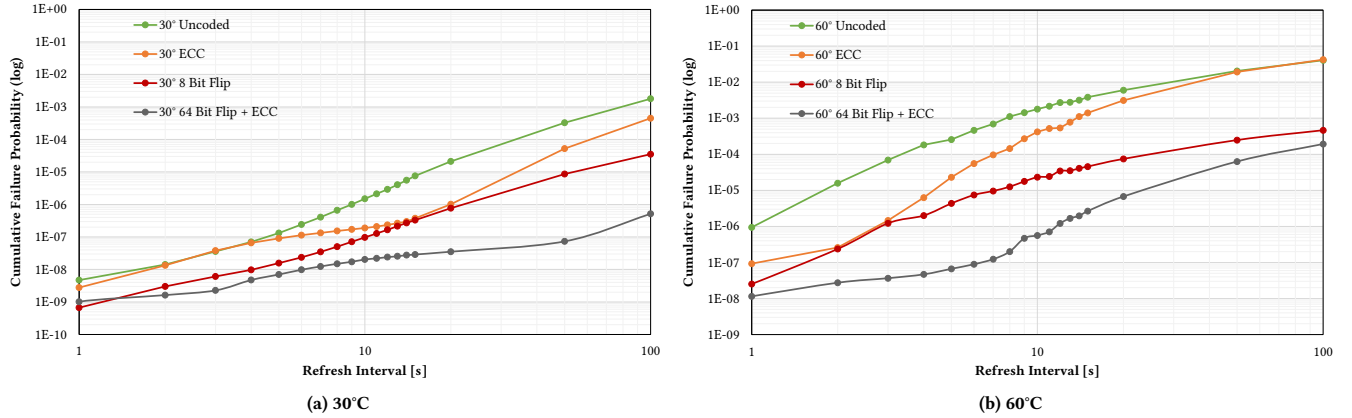


Figure 9: 8 Bit Dynamic Range

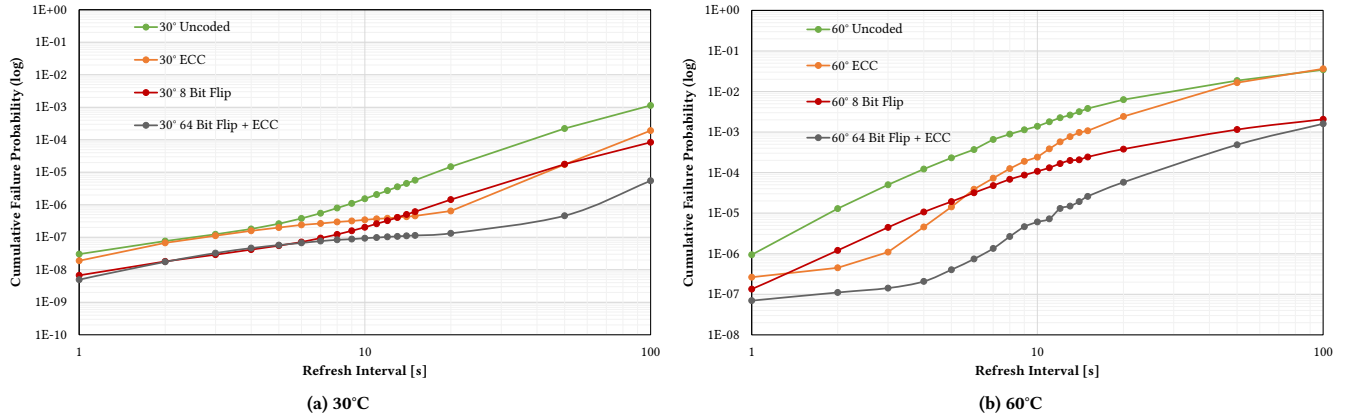


Figure 10: 16 Bit Dynamic Range

9 CONCLUSION

In this paper, a refined model for DRAM errors was presented and verified by various measurements. In addition, it was analyzed from an information theoretic perspective. Based on this model, we presented a coding scheme that uses the DBI mechanism of current DDR4 DRAM devices to store data inverted or not, depending on the data and the characteristics of the DRAM cell (true- or anti-cell). This scheme can significantly lower the error rate for specific applications, and even outperform state of the art ECC techniques.

ACKNOWLEDGMENTS

This work was partially accomplished in cooperation with Huawei and supported by the Carl-Zeiss Stiftung and the project OPRECOMP. 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.

REFERENCES

- [1] A. Bacchini. Characterization of Data Retention Faults in DRAM Devices. 2014.
- [2] S. Baek, S. Cho, and R. Melhem. Refresh now and then. *IEEE Transactions on Computers*, 63(12):3114–3126, 2014.
- [3] J. M. Berger. A note on error detection codes for asymmetric channels. *Information and Control*, 4(1):68–73, 1961.
- [4] I. Bhati, M.-T. Chang, Z. Chishti, S.-L. Lu, and B. Jacob. DRAM Refresh Mechanisms, Trade-offs, and Penalties. *IEEE Transactions on Computers*, PP(99):1–1, 2015.
- [5] T. J. Dell. A white paper on the benefits of chipkill-correct ECC for PC server main memory. *IBM Microelectronics Division*, 11, 1997.
- [6] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950.
- [7] C.-S. Hou, J.-F. Li, C.-Y. Lo, D.-M. Kwai, Y.-F. Chou, and C.-W. Wu. An FPGA-based test platform for analyzing data retention time distribution of DRAMs. In *VLSI Design, Automation, and Test (VLSI-DAT)*, 2013 International Symposium on, pages 1–4, April 2013.
- [8] M.-Y. Hsiao. A class of optimal minimum odd-weight-column SEC-DED codes. *IBM Journal of Research and Development*, 14(4):395–401, 1970.
- [9] IIRDS2017. International Roadmap for Devices and Systems, 2017.
- [10] ITRS 2.0. International Technology Roadmap for Semiconductors, 2015 Edition, Section 5: More Moore.
- [11] JEDEC. JESD79-4B, JEDEC Standard, DDR4 SDRAM. 2017.
- [12] M. Jung, D. Mathew, C. Rheinländer, C. Weis, and N. Wehn. A Platform to Analyze DDR3 DRAM's Power and Retention Time. *IEEE Design & Test*, 34(4):52–59, Aug

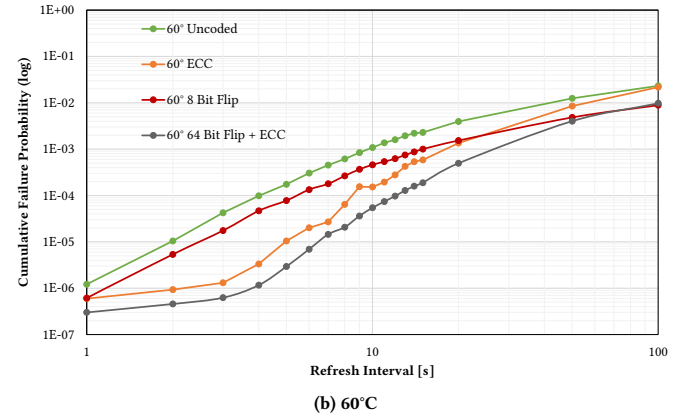
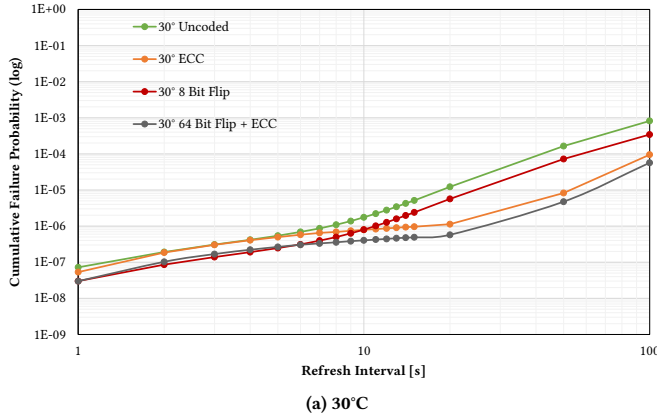


Figure 11: 32 Bit Dynamic Range

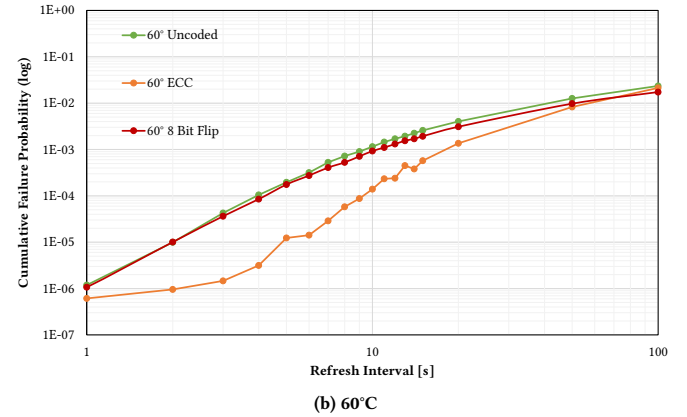
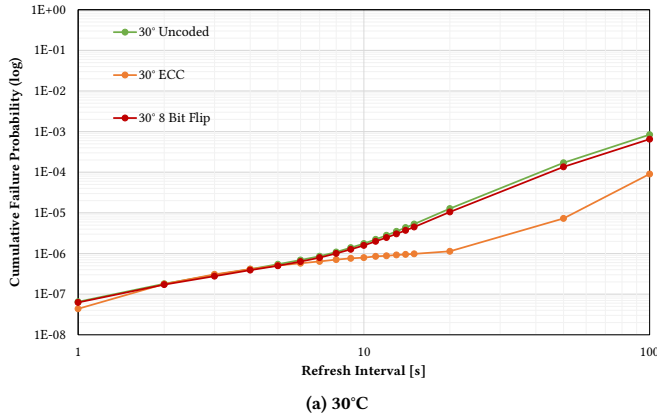


Figure 12: Random Data

- 2017.
- [13] M. Jung, D. M. Mathew, C. Weis, and N. Wehn. Efficient Reliability Management in SoCs - An Approximate DRAM Perspective. In *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016.
 - [14] M. Jung, C. Rheinländer, C. Weis, and N. Wehn. Reverse Engineering of DRAMs: Row Hammer with Crosshair. In *International Symposium on Memory Systems (MEMSYS 2016)*, 2016.
 - [15] Y. Katayama, E. Stuckey, S. Morioka, and Z. Wu. Fault-tolerant refresh power reduction of DRAMs for quasi-nonvolatile data retention. In *Defect and Fault Tolerance in VLSI Systems, 1999. DFT '99. International Symposium on*, pages 311–318, Nov 1999.
 - [16] 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.
 - [17] K. Kraft, M. Jung, C. Sudarshan, D. M. Mathew, C. Weis, and N. Wehn. Improving the Error Behavior of DRAM by Exploiting its Z-Channel Property. In *IEEE Conference Design, Automation and Test in Europe (DATE)*, March 2018.
 - [18] H. J. Kwon, E. Seo, C. Y. Lee, Y. H. Seo, G. H. Han, H. R. Kim, J. H. Lee, M. S. Jang, S. G. Do, S. H. Cho, J. K. Park, S. Y. Doo, J. B. Shin, S. H. Jung, H. J. Kim, I. H. Im, B. R. Cho, J. W. Lee, J. Y. Lee, K. H. Yu, H. K. Kim, C. H. Jeon, H. S. Park, S. S. Kim, S. H. Lee, J. W. Park, S. S. Lee, B. T. Lim, J. y. Park, Y. S. Park, H. J. Kwon, S. J. Bae, J. H. Choi, K. I. Park, S. J. Jang, and G. Y. Jin. 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.
 - [19] C. K. Lee, Y. J. Eom, J. H. Park, J. Lee, H. R. Kim, K. Kim, Y. Choi, H. J. Chang, J. Kim, J. M. Bang, S. Shin, H. Park, S. Park, Y. R. Choi, H. Lee, K. H. Jeon, J. Y. Lee, H. J. Ahn, K. H. Kim, J. S. Kim, S. Chang, H. R. Hwang, D. Kim, Y. H. Yoon, S. H. Hyun, J. Y. Park, Y. G. Song, Y. S. Park, H. J. Kwon, S. J. Bae, T. Y. Oh, I. D. Song, Y. C. Bae, J. H. Choi, K. I. Park, S. J. Jang, and G. Y. Jin. 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.
 - [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 *IEEE 30th International Conference on Computer Design (ICCD)*, 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.-L. Lu. Tutorial: DRAM - Circuits, Organization, Interfaces. In *IEEE Microarchitecture Conference (MICRO)*, 2016.
 - [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] D. M. Mathew, M. Schultheis, C. C. Rheinländer, C. Sudarshan, M. Jung, C. Weis, and N. Wehn. An Analysis on Retention Error Behavior and Power Consumption

- of Recent DDR4 DRAMs. In *IEEE Conference Design, Automation and Test in Europe (DATE)*, 2018.
- [26] N. J. MIN HEE CHO. An Innovative Indicator to Evaluate DRAM Cell Transistor Leakage Current Distribution. *Journal of the Electron Devices Society*, 2017.
 - [27] P. J. Nair, D.-H. Kim, and M. K. Qureshi. ArchShield: Architectural Framework for Assisting DRAM Scaling by Tolerating High Error Rates. In *Proceedings of the 40th Annual International Symposium on Computer Architecture, ISCA '13*, pages 72–83, New York, NY, USA, 2013. ACM.
 - [28] S.-K. Park. Technology scaling challenge and future prospects of DRAM and NAND flash memory. 2015.
 - [29] M. Patel, J. S. Kim, and O. Mutlu. The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017.
 - [30] K. I. Santosh K. Kurinec, editor. *Nanoscale Semiconductor Memories: Technology and applications*. CRC Press, 2013.
 - [31] T. Schloesser. 6F2 buried wordline DRAM cell for 40nm and beyond. *IEEE International Electron Devices Meeting, San Francisco, CA*, 2008.
 - [32] S. M. Seyedzadeh. Mitigating Bitline Crosstalk Noise in DRAM Memories. 2017.
 - [33] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July–Oct. 1948.
 - [34] L. G. Tallini, S. Al-Bassam, and B. Bose. On the capacity and codes for the Z-channel. In *IEEE International Symposium on Information Theory*, page 422. IEEE, 2002.
 - [35] M. Technology. Using DDR4 in Networking Subsystems. Technical report, 2014.
 - [36] C. S. Thomas To. Optimal DDR4 System with Data Bus Inversion (DBI). In *Design Con*, 2016.
 - [37] J. Wang, X. Dong, and Y. Xie. ProactiveDRAM: A DRAM-initiated retention management scheme. In *32nd IEEE International Conference on Computer Design (ICCD)*, pages 22–27, Oct 2014.
 - [38] 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 IEEE Conference on Design, Automation & Test in Europe (DATE)*. European Design and Automation Association, 2015.
 - [39] Z. Yang and S. Mourad. Crosstalk in Deep Submicron DRAMs. 2000.
 - [40] Y. Zemo and M. Samisha. Crosstalk Induced Fault Analysis and Test in DRAMs. 2006.