

Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, *Hongzhong Zheng,
**John Halbert, **Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel*

Abstract— Today's DRAM process is expected to continue scaling, enabling minimum feature sizes below 10nm. To achieve this, the main challenges to address are expected to be refresh, write recovery time (tWR), and variable retention time (VRT) parameters. This paper proposes enhancement features that address these three scaling parameters by simultaneously co-architecting the controller and DRAM instead of designing them individually. Combining temperature compensated tWR (TCWR) with sub-array level parallelism (SALP) enables tWR relaxation, while further improving performance above a given temperature threshold such as 25°C. In-DRAM ECC enables efficient repairs when refresh and VRT failures generate fail bits. It also provides additional single bit failure protection, thereby increasing field reliability which is critical for servers. Despite the general perception that in-DRAM ECC is not applicable to server main memories due to excessive chip size increases, the proposed in-DRAM ECC with dummy data bit pre-fetching results in a modest array size increase of only ~6.2%. By changing the ecosystem to enable the proposed features, DRAM scaling is expected to accelerate further, enabling a continuous supply of low cost, high capacity, high performance, and low power DRAMs for the industry.

Index Terms—DRAM process scaling, refresh, write recovery time (tWR), variable retention time (VRT), sub-array level parallelism (SALP), temperature compensated tWR (TCWR), parameter relaxation, sub-array, page buffer, in-DRAM ECC, repair, redundancy



1 INTRODUCTION

During the past few decades, high density, high speed, and low power memory system requirements have grown continuously. DRAM technology has evolved over the years to satisfy such needs. The key enabler of such evolution has been continuous scaling of physical feature size from one DRAM process generation to the next. DRAM scaling also enables continued bit-cost reduction which is another important driving factor in the DRAM industry.

Such requirements are expected to continue increasing, and DRAM process scaling will continue beyond 10nm to meet such expectations. In these efforts, the main challenging parameters to address are expected to be refresh, write recovery time (tWR), and variable retention time (VRT). By introducing new DRAM scaling features to address each parameter, it is expected that DRAM process scaling will accelerate even faster in the future.

This paper proposes new approaches to enhance DRAM process scaling by co-architecting the controller and DRAM together instead of designing them individually. Section 2 describes in detail the three expected process-scaling parameter challenges - refresh, tWR and VRT - and their contributing factors. Section 3 introduces three scaling enhancing features to overcome these challenges.

2 DRAM SCALING CHALLENGES

The main expected process-scaling challenges are expected to be refresh, tWR, and VRT. They are described in

section 2.1, 2.2, and 2.3, respectively.

2.1 Refresh

Since DRAM is a volatile memory, it is essential to re-charge stored charges periodically. Otherwise, charges are lost over time. DRAM cell retention time (t_{RET}) is determined by the time it takes for charges to leak out from the full voltage level (V_{DD}) down to the minimum voltage level that can be sensed by the page buffer or bit-line sense amplifier (V_{min}). t_{RET} can be calculated as shown below where I_{leak} and C_s respectively represent cell transistor leakage current and cell capacitance. As shown, it is critical to keep the cell capacitance large and leakage current small.

$$t_{RET} = \int_{V_{min}}^{V_{DD}} \frac{C_s \cdot dV_c}{I_{leak}(V_c)} \approx \frac{C_s \cdot (V_{DD} - V_{min})}{I_{leak}}$$

However, as scaling reduces cell feature sizes, higher aspect ratio cell capacitors are required to maintain the same cell capacitance as in previous process generations. However, building high-aspect ratio cell capacitors is a challenging process technology, potentially putting limitations on the maximum achievable DRAM cell capacitance. This results in a cell capacitance decrease with scaling, resulting in a higher probability of sense amplifier operation failures. Also, keeping the leakage current at the same level, while shrinking the cell feature size, is becoming increasingly challenging, resulting in larger charge loss for a given refresh period.

2.2 Write Recovery Time

Write recovery time (tWR) is the time it takes to write data into a DRAM cell. It represents the time it takes from the last write data burst to the precharge command issued to the same bank. As scaling continues, this timing requirement is becoming difficult to satisfy since related parasitic DRAM core components become worse. Figure 1 shows the write data path which consists of the global IO (GIO) driver, column select line (CSL) transistor, bit-line sense amplifier, and the access transistor.

As transistors scale down, the time required to charge the storage capacitor fully is expected to increase. The main factors include contact resistance between the cell capacitor and the access transistor, on-current of the access transistor, and bit-line resistances.

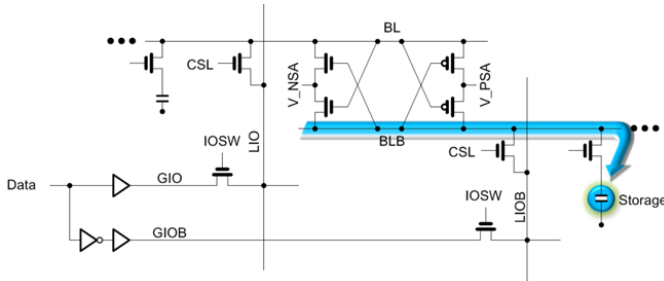


Fig. 1. Critical data path to write data into the DRAM cell storage.

2.3 Variable Retention Time

DRAMs have always exhibited variable retention time (VRT) phenomena. Currently, there are no efficient ways of fundamentally pre-screening VRT bits during production testing. So far, most manufacturers have been able to manage it by increasing average retention time and by enforcing larger test screen margin. With future process scaling, VRT is expected to become more challenging since such occurrences may become more frequent as cell capacitance decreases.

With VRT failure bits, the retention time changes randomly between two or three distinct states. It is believed to be related to the random telegraph noise (RTN) in gate-induced drain leakage (GIDL) current at the storage node [1]. The Si/SiO₂ interface charge trap, caused by the Si dangling bonds, is believed to generate GIDL current fluctuations [2]. Because of such random VRT characteristics, managing VRT failures is very difficult. The most effective way to deal with VRT is generally believed to be ECC.

3 DRAM SCALING ENHANCING FEATURES

To address the expected scaling parameter challenges Section 2 described, DRAM AC parameters, including refresh and tWR, can be relaxed. Alternately, new area efficient cell repair mechanisms can be introduced. The following proposes scaling enhancing features to address each of these methods. Section 3.1 and Section 3.2 describe sub-array level parallelism and temperature com-

pensated tWR (TCWR) which correspond to the former case, compensating performance after relaxing tWR. It is also shown that performance can be further improved by combining SALP and TCWR. Section 3.3 introduces in-DRAM ECC with dummy data pre-fetch which corresponds to the latter case intended to repair single-bit failures, including VRT and refresh errors, in a more efficient manner compared to traditional row and column redundancy schemes.

3.1 Sub-array Level Parallelism

One way of enhancing DRAM scaling is to extend tWR. Figure 2 shows fail bit counts due to tWR failures for different process nodes. 2x/2y/2z and 1x/1y/1z in the x-axis respectively represent the largest/mid/lowest minimum feature sizes within 20nm and 10nm process nodes. In this figure, fail bit count data for the 2x case is measured, while all others are predicted through simulation. By relaxing tWR by a factor of 5 (from 15ns to 75ns) it is seen that fail bit counts are reduced by orders of magnitude. However, relaxing tWR impacts performance. To compensate for such performance loss requires new compensating features.

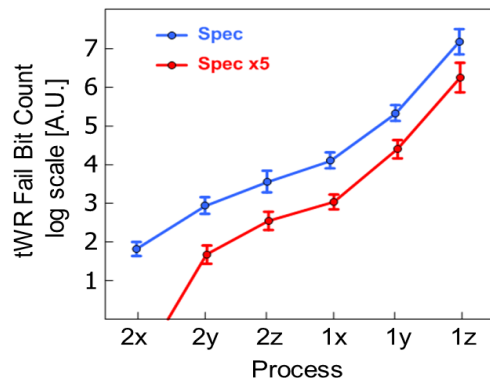


Fig. 2. Fail bit count estimation with and without tWR relaxation for different DRAM process nodes.

For such a purpose, sub-array level parallelism was previously proposed [3]. The idea here is to utilize a DRAM bank's multiple page buffers. When a page is opened in a bank and the next activate request comes to a different sub-array in the same bank, the second requested page can be opened in parallel.

Figure 3(a) shows timing diagrams for normal and SALP cases. In the normal case, the next active command to the same bank can come only after tRAS+tRP. On the other hand, in case of SALP, the next active command to a different sub-array can come after tRAS. As a result, the next activate command can be pulled in by tRP compared to the normal case. To simplify this operation, a mutually exclusive sub-array activate command is proposed where all other pages in other sub-arrays are closed once a new activate request to a closed sub-array page is made.

Figure 3(b) shows timing diagrams for the write normal and SALP cases. In the normal case, the next activate

command can be issued only after the current bank is fully pre-charged. However, in the SALP case, the next activate request to a different sub-array can come earlier by $tRP+tWR-tWA$. tWA is a new parameter which needs to be introduced to prevent data being written to the current page, from overwriting data in the next activated sub-array.

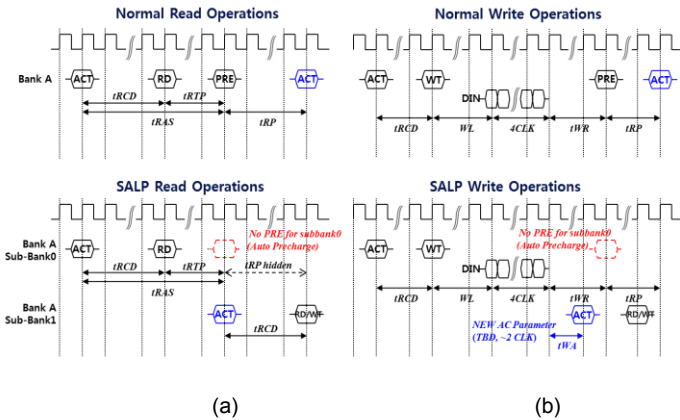


Fig. 3. (a) Read and (b) write timing diagrams for normal and SALP cases.

Figure 4 shows system performance simulation results for various workloads when tWR is relaxed by a factor of 2 and 3, and when SALP is applied with 2 sub-arrays per bank. Results show that the performance is reduced by ~5% and ~2% in average if tWR is relaxed by a factor of 3 and 2, respectively. It also shows that the performance is compensated, and even improved, to up to ~3% in average when SALP is applied - even with tWR relaxed by a factor of 3.

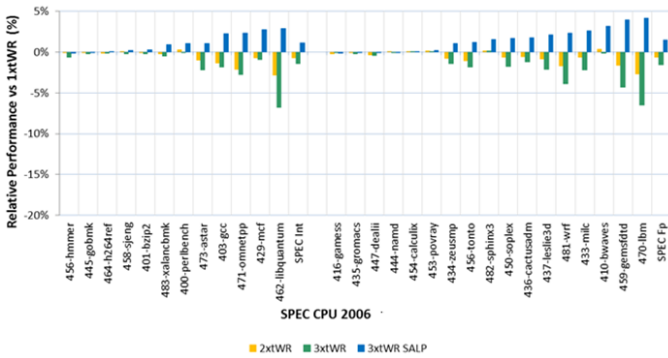


Fig. 4. Performance simulation results when tWR is relaxed by a factor of 2 and 3, and when SALP is applied to compensate for the performance loss.

3.2 Temperature Compensated tWR

In case of SALP, tWR is relaxed for all temperature ranges. However, tWR is known to be worse at cold temperature. When combined with SALP, having different tWR specifications for different temperature ranges (longer at cold temperature) can enable performance gains at higher temperatures. This also allows near-

equivalent DRAM yield gain as when tWR is relaxed for the entire temperature range, since tWR fail bits are mostly determined at cold temperatures.

For example, tWR can have the same value as the current spec (15ns) above a certain temperature threshold such as 25°C, while it can be relaxed to a larger value (ex. 45ns) below the temperature threshold as shown in Fig.5. In such cases, performance improves because of SALP when ambient system operation temperatures exceed the temperature threshold most of the time. The controller will periodically check each DRAM's temperature through a predefined protocol to adjust tWR depending on the temperature information provided.

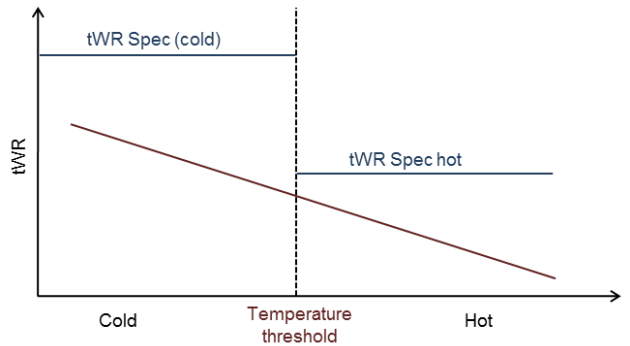


Fig. 5. tWR value trend with change in temperature and corresponding tWR specification (tWR worse at cold temperature).

3.3 In-DRAM ECC with Dummy Data Pre-fetching

In-DRAM ECC can repair single fail bits and is known to have high repair efficiency compared to traditional row and column repair redundancy schemes. For such reasons, the LPDDR4 specification has already adopted some means to enable in-DRAM ECC.

In main memories for servers, it has been generally believed it is difficult to apply in-DRAM ECC since X4 DRAMs are mostly used in order to support high capacity modules. In both DDR3 and DDR4, each X4 DRAM reads and writes data in 8-bit bursts, resulting in a 32-bit data access granularity. Protecting 32-bits by SECDED ECC requires an additional 6 parity bits, resulting in increasing DRAM cell array size approximately 20% which is economically impractical.

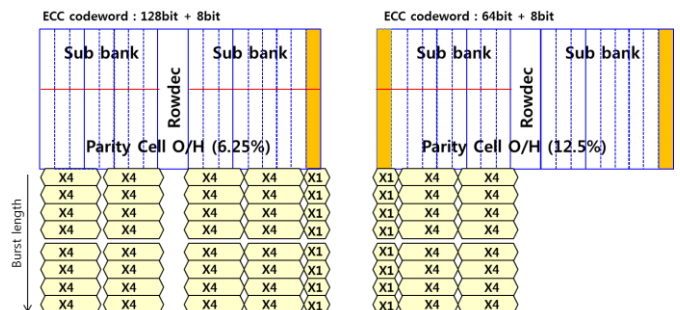


Fig. 6. Proposed extra dummy data bit pre-fetching (64 and 128bit cases) for reducing DRAM cell array size overhead.

To overcome such limitations, a dummy data bit pre-fetching scheme is proposed. As Fig.6 shows, the main concept is to pre-fetch extra bits inside the DRAM even though only 32-bits are actually required for reads and writes. Fig. 6 gives 64-bit and 128-bit dummy pre-fetch example cases. For these example cases, DRAM cell array size overheads are 12.5% and 6.25% respectively. Extra data can be read or written by enabling more column select lines inside the column address decoder.

However, as a result of dummy data bit pre-fetching, all writes need to be replaced by read-modify-write because only parts of the data code words are overwritten. Hence, parity bits need updating for every write. Due to such requirements, tCCD_L_WR (tCCD for write within same bank group in DDR4) needs to be increased from 6~9tck to 25~29tck. As Fig. 7 shows, a write command internally first initiates a read to the given column address, followed by data writes. Therefore, tWTR_L (write to read) needs to be observed from the last write-data burst to the next write command within the same bank group, since the next write is actually an internal read.

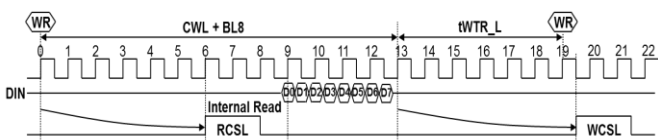


Fig. 7. Write to write timing diagram within the same bank group in case of read-modify write.

To estimate system performance impact from increasing tCCD_L_WR, performance simulations were run for different workloads. Figure 8 shows performance estimation results when tWR is relaxed by a factor of 3, tCCD_L_WR is increased to 24tck, and SALP is used to compensate for the performance loss. Simulation conditions are tCCD_L_WR=24tck, tWR 3X, DDR4-2400, 6 channels, 1 DIMM per channel, and 2 rank per DIMM. Results show an average performance loss of approximately 2%. However, by applying in-DRAM ECC only, without increasing tWR, performance loss is expected to break even when applying SALP.

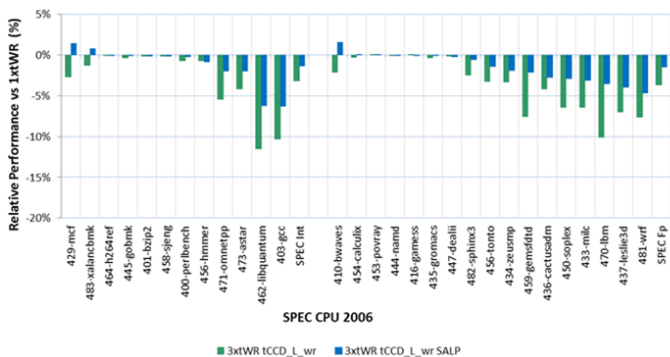


Fig. 8. Performance simulation results with 3X tWR relaxation, tCCD_L_WR increase to 24tck, and with SALP to compensate for the performance loss.

Another benefit of having in-DRAM ECC is field reliability improvement. Figure 9 shows failure rate probability analysis results with, and without, in-DRAM ECC, and when SECDED or SDDC (chip-kill) is applied at the DIMM level. It has been assumed the DIMM consists of 4Gb X4 DRAMs. Also, it has been assumed all fail bits are single bits, and they are randomly scattered across the entire DIMM. The x-axis is the absolute bit error rate present in the DIMM, while the y-axis represents the bit error rate after applying the corresponding module and DRAM-level ECC combinations. Results show that the bit error rate decreases by orders of magnitude when in-DRAM ECC is adopted.

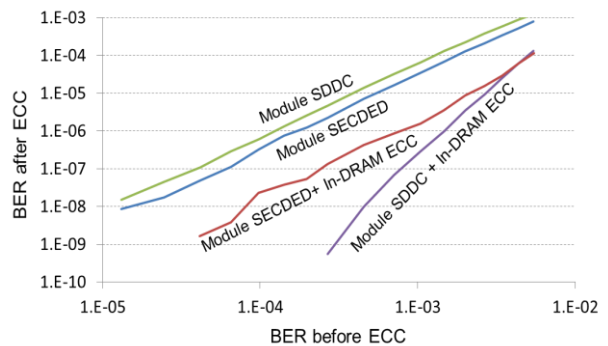


Fig. 9. DRAM bit failure rate probability analysis results with, and without, in-DRAM ECC, and when SECDED or SDDC is applied at the DIMM level.

5 CONCLUSION

To enhance DRAM process scaling, various challenging process parameters, including refresh, tWR, and VRT, need to be addressed. Combining SALP with TCWR enables tWR relaxation by compensating for the performance loss caused by this parameter relaxation. It can even improve performance above given temperature thresholds such as 25°C.

In-DRAM ECC allows efficient repair of fail bits coping with VRT and refresh failures. It also provides extra protection from single bit failures, resulting in increased field reliability. By enabling these features by co-architecting controllers and DRAM, future DRAM process scaling is expected to accelerate further beyond 10nm.

REFERENCES

- [1] D. S. Yaney, C. Y. Lu, R. A. Kohler, M. J. Kelly, and J. T. Nelson, "A Meta-stable Leakage Phenomenon in DRAM Charge Storage - Variable Hold Time," *IEDM Technical Digest*, pp. 336-339, 1987.
- [2] Y. Mori, K. Ohyu, K. Okonogi and R. Yamada, "The Origin of Variable Retention Time in DRAM," *IEDM Technical Digest*, pp.1034-1037, 2005.
- [3] Y. Kim, V. Seshadri, D. Lee, J. Liu, and O. Mutlu, "A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM," *Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2012.