DRAMScale: Mechanisms to Increase DRAM Capacity

Krishna T. Malladi Uksong Kang Manu Awasthi Hongzhong Zheng
Samsung Semiconductor, Inc.
{k.tej, uksong.kang, manu.awasthi, hz.zheng}@samsung.com

ABSTRACT

New resistive memory technologies promise scalability and non-volatility but suffer from longer, asymmetric read-write latencies and lower endurance, placing the burden of system design on architects. In order to avoid such pitfalls and still provision for exascale data requirements using a much faster DRAM technology, we introduce DRAMScale. It features three novel mechanisms to increase DRAM density while complementing technology scaling and creating a new capacity-optimized DRAM system. Such optimizations enable us to build a two-tier memory system that meets memory latency and capacity requirements.

CCS Concepts

•Hardware → Dynamic memory; •Computer systems organization → *Cloud computing;*

Keywords

Big Data, DRAM, Memory, Capacity, Latency, Tiered

1. INTRODUCTION

A large number of present day "Big Data"applications are increasingly becoming memory bound [2, 1, 5]. Applications and frameworks are being designed to contain entire datasets in main memory, possibly across multiple nodes on a network [7]. These applications aim for faster data processing by reducing data access time and are driving innovations in low-latency, high capacity memory architectures.

Non-volatile memory technologies such as PCM, STTM-RAM and ReRAM are on the horizon promising non-volatility and technology scalability. However, DRAM continues to have significant advantages in latency and endurance. Furthermore, DRAM technology continues to scale very efficiently and is enabling new and better memory systems [3]. However, the data sizes continue to increase, making bigger demands to migrate more data to main memory from storage. For example, Apache Spark places datasets in memory for fast analytics. However, the Spark platform still optimizes for data spillover to the storage subsystem which is often 100-1000× slower [7]. Clearly, this gap between memory and storage capacities needs to be bridged.

In this paper, we introduce **DRAMScale**, which comprises of three DRAM level innovations to increase DRAM density and assist in technology scaling. This allows for the creation of a new, high-capacity DRAM tier by trading off high speed DRAM interfaces to increase areal density.

MEMSYS 2016 October 3–6, 2016, Washington, DC, USA

ACM ISBN 978-1-4503-4305-3.

DOI: http://dx.doi.org/10.1145/2989081.2989109

2. TIERING MAIN MEMORY

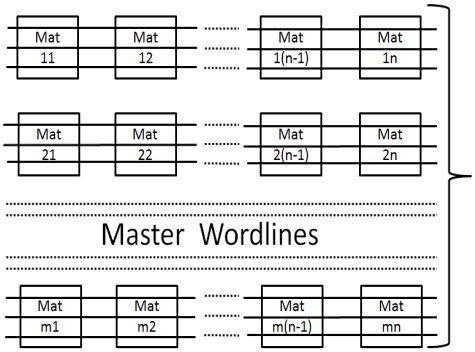
In this section, we describe the details of DRAMScale that enable high capacity memory tier which is much faster than existing storage systems. DRAM chip area consists of memory cell area, the inter-cell spacing and peripheral circuits that enable access to the dense chips. Prior work shows that up to 10% of the chip area is invested in local, sub-wordline drivers while up to 15% is occupied by local Bitline Sense Amplifiers (BLSA). Another large fraction is occupied by high-speed metal routing for datalines [8, 4]. These peripheral circuits help achieve low latency (tAC) in three main activities i.e. row activation, sense amplification and internal data movement. DRAMScale optimizes across the above systems to cumulatively save chip area, at the cost of increased DRAM access latency. With its flexible design choices, DRAMScale provides a new DRAM-based intermediate capacity tier, between main memory and storage.

2.1 Sub-wordline drivers

DRAM devices are organized as multiple banks, with each bank comprising multiple arrays. The number of arrays depends on the chip device output size. Each array is again divided into a grid of sub-arrays or mats with multiple rows and columns with DRAM cells [6, 4]. Multiple sub-arrays on a row share a global wordline that performs DRAM row activation. In order to ease load on this wordline, a hierarchical scheme distributes the wordline function to a sub-wordline driver driving each sub-array. This driver buffers and connects cells in a row of that sub-array. Since there are many sub-arrays in a multi-gigabit device, sub-wordline drivers occupy significant fraction of chip area.

DRAMScale's **first optimization** eliminates the local sub-wordline drivers and instead uses the master wordlines to drive all the DRAM rows across multiple sub-arrays, as shown in Figure 1. While this increases the resistive load, it is still feasible to activate the wordline within a reasonable latency, depending on the load on each master wordline. This presents a flexible trade-off between the area and the

Figure 1: Master wordlines across sub-arrays



row activation time that could be tuned to meet design constraints. The latency spectrum between the memory and storage is 1000×, leaving ample room for optimizations.

2.2 Local bitline sense amplifiers

The next optimization is focused on the BLSAs in the sense amplification path. After word line activation, DRAM cells share cell charge with their corresponding bitlines. BLSA amplifies this charge by increasing the voltage difference across bitline (BL) and its complement $\overline{BL}$. In order to perform this efficiently, a hierarchical scheme is employed, in which each subarray is provided with a local BLSA. The amplified voltage difference is relayed to the global datalines and the data is also restored to the DRAM cells as DRAM reads are destructive. Consequently, a precharge operation restores bitlines to their original level. These latencies are affected by the access speed of BLSAs. However, these occupy a large fraction of the overall chip area, because of both their size and quantity.

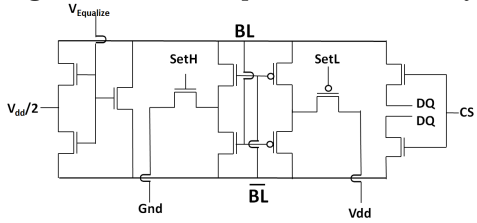
One way to improve the cumulative BLSA area is to reduce their quantity and have one BLSA share the load from more DRAM subarrays/cells, similar to DRAMScale's wordline optimization. However, as DRAM capacitors scale, the charge differential between the BL and $\overline{BL}$ is small. Larger BL load reduces this further and will decrease DRAM access reliability, making the approach unviable. So, we instead focus on BLSA area efficiency for our **second optimization**.

As shown in Figure 2, a typical BLSA consists of a few CMOS transistors. We propose to reduce the transistor widths (W) for all the BLSAs by a flexible amount to provide a continuous latency-capacity trade-off curve. This increases the DRAM access latency since smaller transistors increase resistivity. However, this significantly reduces the area of BLSA which is one of the most repeated blocks in a multi-gigabit chip. An important precaution is to preserve the relative transistor sizes so as to maintain offset differences between BL and $\overline{BL}$. This enables the BLSA to continue to be immune to circuit noise, similar to the baseline.

2.3 Routing

We finally focus on the data movement from bitlines to the chip's DQ output. DRAM DQ paths are optimized for latency by using metal for the routing layers. While such routing schemes enable high speed data access, they also occupy area because of metal pitch spacing constraints. DRAMScale's **third optimization** increases DRAM area efficiency by using poly-silicon or other engineered materials for higher chip density. Specifically, we focus on DRAM local I/O lines in the subarray, global I/O lines from BLSAs that are routed in metal to change to new routing materials. This saves area and also inter-spacing constraints. We could also use poly-silicon for the DRAM DQ datapath metal bus at the boundary of global I/O sense amplifier where global data lines meet and interface with the DRAM module.

Figure 2: Sense amplification circuitry



2.4 System Integration

As discussed, both memory latency and capacity play an important role in meeting application requirements. To meet these, we deploy high performance regular DRAM and DRAMScale optimized DRAM in a tiered architecture. OS could present this system either via hardware caching or software tiering. In the former, the regular DRAM tier acts as a hardware cache, with data placement decisions made within hardware controllers. In the latter option, OS addresses both the tiers' capacity, while placing latency sensitive pages in the latency tier. Overall, DRAMScale tier promotes more data migration from the storage layer and assists memory technology scaling to meet future capacity and latency needs.

3. CONCLUSION

In this paper, we present DRAMScale, a set of DRAM level optimizations that assist in increasing chip density to build a DRAM-only high capacity, low latency architecture. We achieve this by using three innovations to increase the capacity of contemporary DRAM. These include (i) removing the sub-wordline drivers, (ii) reducing the size of the local bitline sense amplifiers, and (iii) proposing to use low area materials like poly-silicon for routing. These three approaches used together can help architect DRAM with larger areal density. Finally, we present mechanisms to integrate DRAMScale into server platforms to present a DRAM-based low latency, high capacity, tiered memory solution.

4. REFERENCES

[1] M. Awasthi. Rethinking Design Metrics for Datacenter DRAM. In *MEMSYS*, 2015.

[2] M. Awasthi et al. System-Level Characterization of Datacenter Applications. In *ICPE*, 2015.

[3] K. Kim. Silicon technologies and solutions for the data-driven world. In *ISSCC*, 2015.

[4] Y. Kim et al. A case for exploiting subarray-level parallelism (salp) in dram. In *ISCA*, 2012.

[5] K. T. Malladi et al. Rethinking DRAM Power Modes for Energy Proportionality. In *MICRO*, 2012.

[6] Y. H. Son et al. Reducing memory access latency with asymmetric DRAM bank organizations. In *ISCA*, 2013.

[7] M. Zaharia et al. Spark: Cluster Computing with Working Sets. In *HotCloud*, 2010.

[8] T. Zhang et al. Half-DRAM: A high-bandwidth and low-power DRAM architecture from the rethinking of fine-grained activation. In *ISCA*, 2014.