Sparkk: Quality-Scalable Approximate Storage in DRAM

Jan Lucas, Mauricio Alvarez-Mesa, Michael Andersch and Ben Juurlink

Embedded Systems Architectures Group - TU Berlin

June 14, 2014

Introduction

- DRAM bitcells are leaky and need refreshing to keep the data
- Most bitcells can keep data for much longer time than the most leaky cells

- Not all data needs bit exact storage
- We can save power and/or gain performance by providing good enough storage instead.

Motivation

- Refresh is projected to use almost half of the DRAM power within just the next 3 device generations![1]
  - Capacity increase $\rightarrow$ more cells need refresh
  - Smaller bit cells $\rightarrow$ cells need to be refreshed more often
- Mobile devices can put CPUs and GPUs into sleep but still need continuous DRAM refresh
- Large parts of the stored data can tolerate imperfect storage

[1] Liu et al.: "RAIDR: Retention-Aware Intelligent DRAM Refresh" ISCA '2012

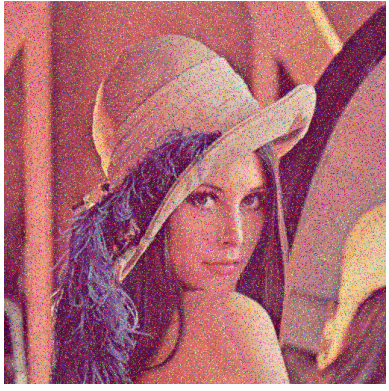
Flikker

Liu et al.: "Flikker: Saving DRAM Refresh-power through Critical Data Partitionin" ASPLOS '2011

- Reduce refresh in one part of the memory
- Provide more power efficient storage
- But only for "non-critical" data.
- Images, sounds, etc.

- Good idea, but we can do better! $\rightarrow$ Let's see how!

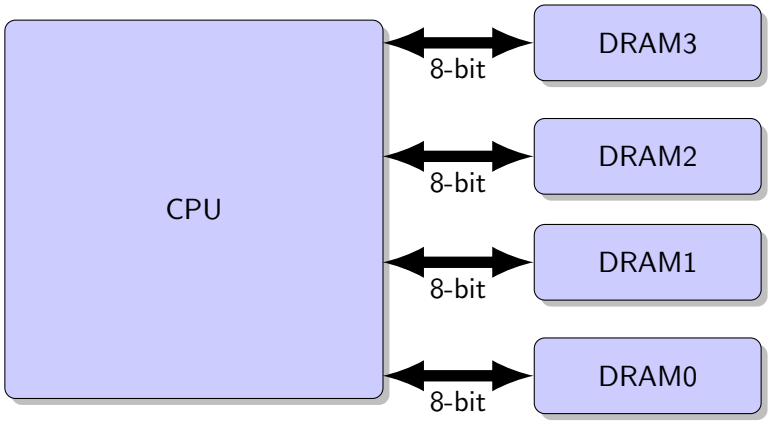
Different significances



10% of all MSBs flipped

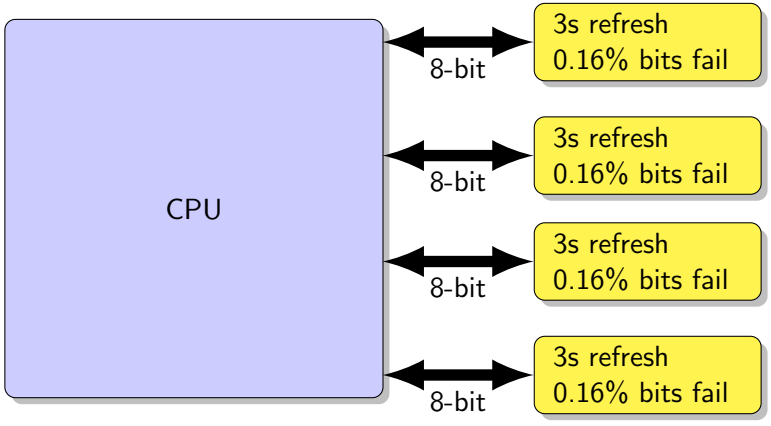
10% of all LSBs flipped

Most DRAM Interfaces use multiple chips



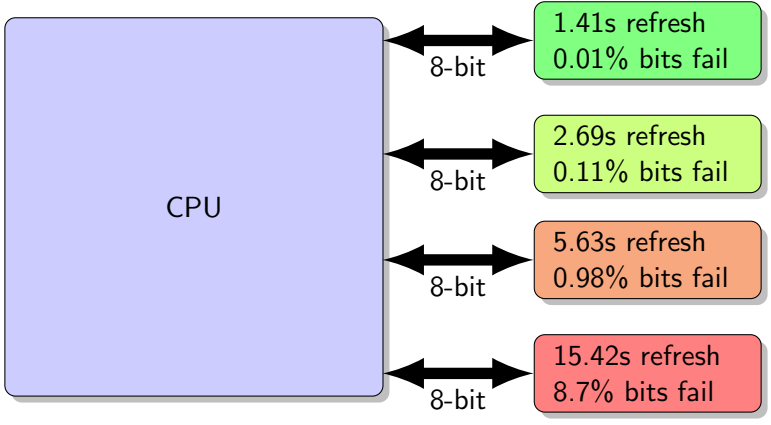
Most DRAM Interfaces use multiple chips

Flikker

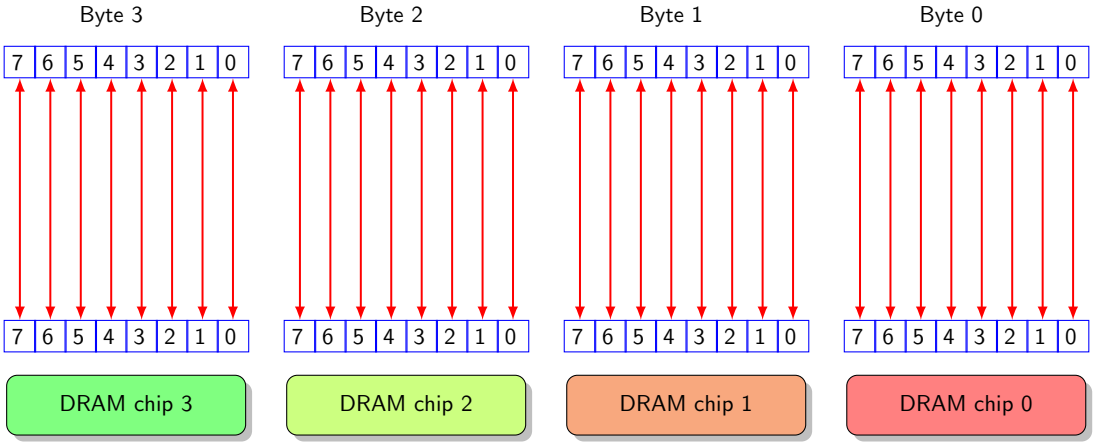


Use different refresh rates on the different chips!

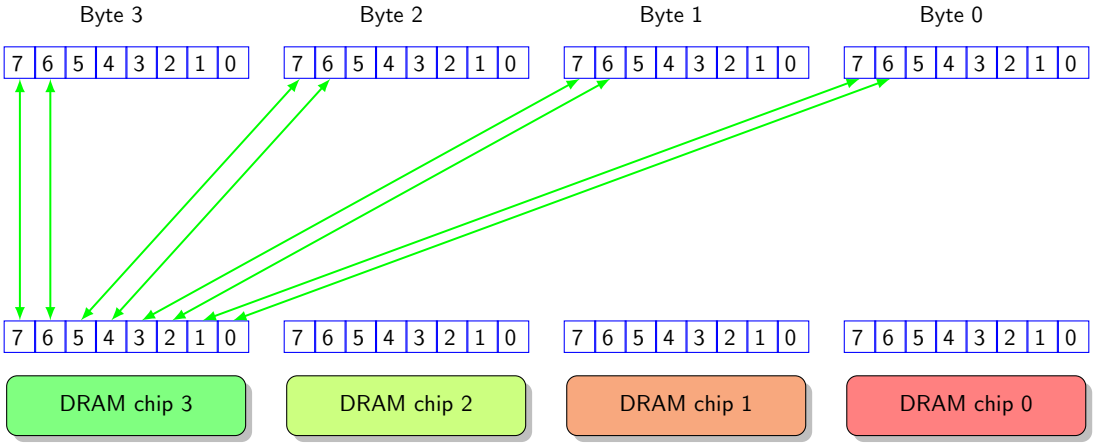
Sparkk



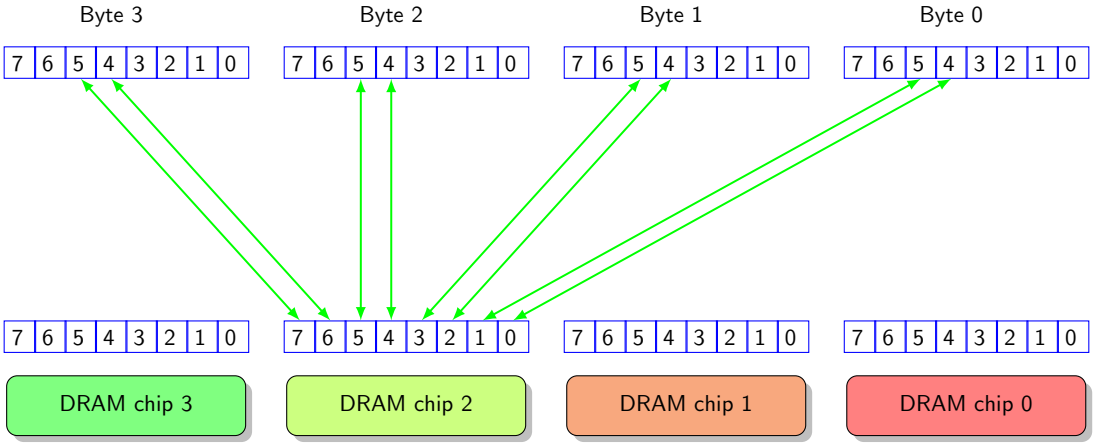
Conventional Mapping of Bytes to DRAM



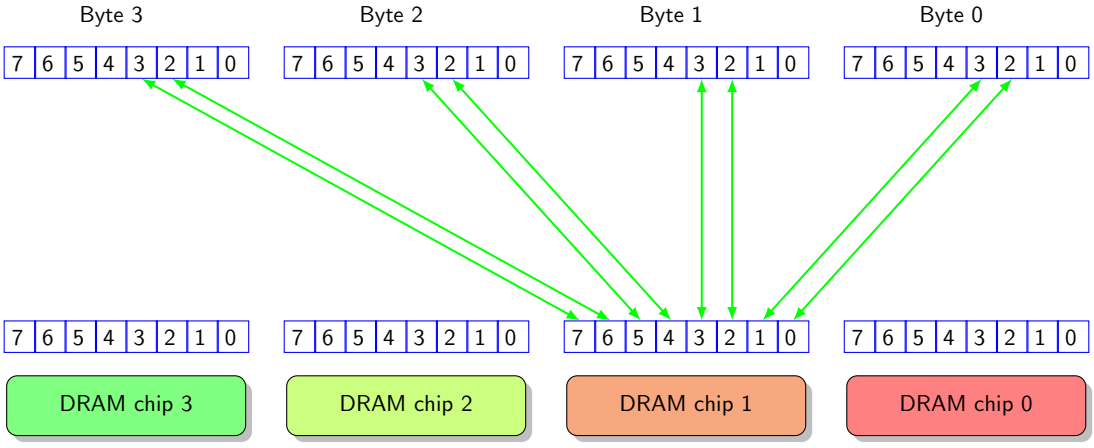
Permutation



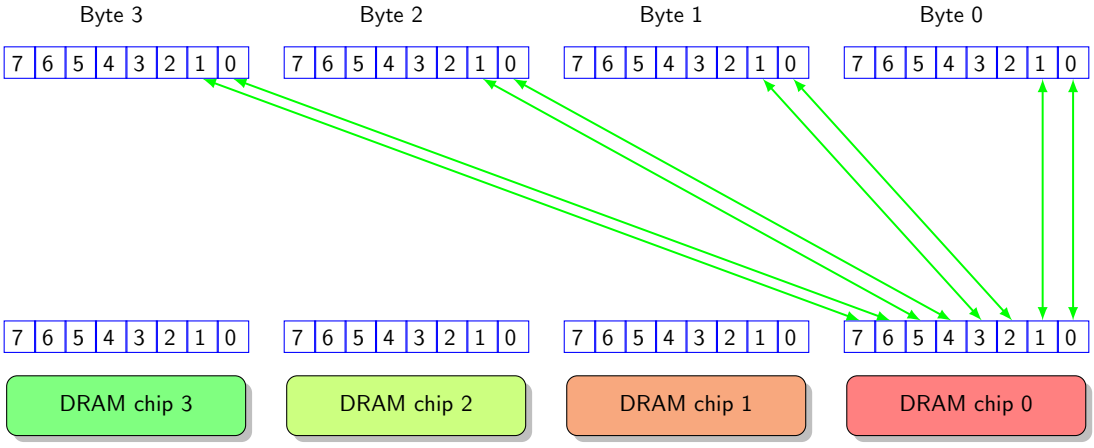
Permutation



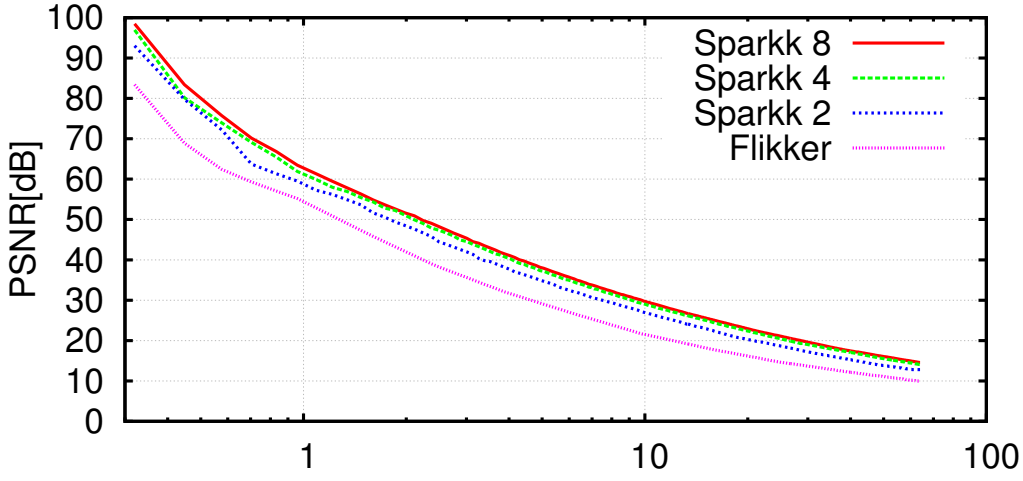
Permutation



Permutation



Quality Results



Comparison

Flikker(8s)

Sparkk(8s)

More information in the paper!

- Choosing optimal refresh rates for the different memory chips
- How to refresh different chips from the same rank at different rates
- Managing memory areas with different refresh requirements