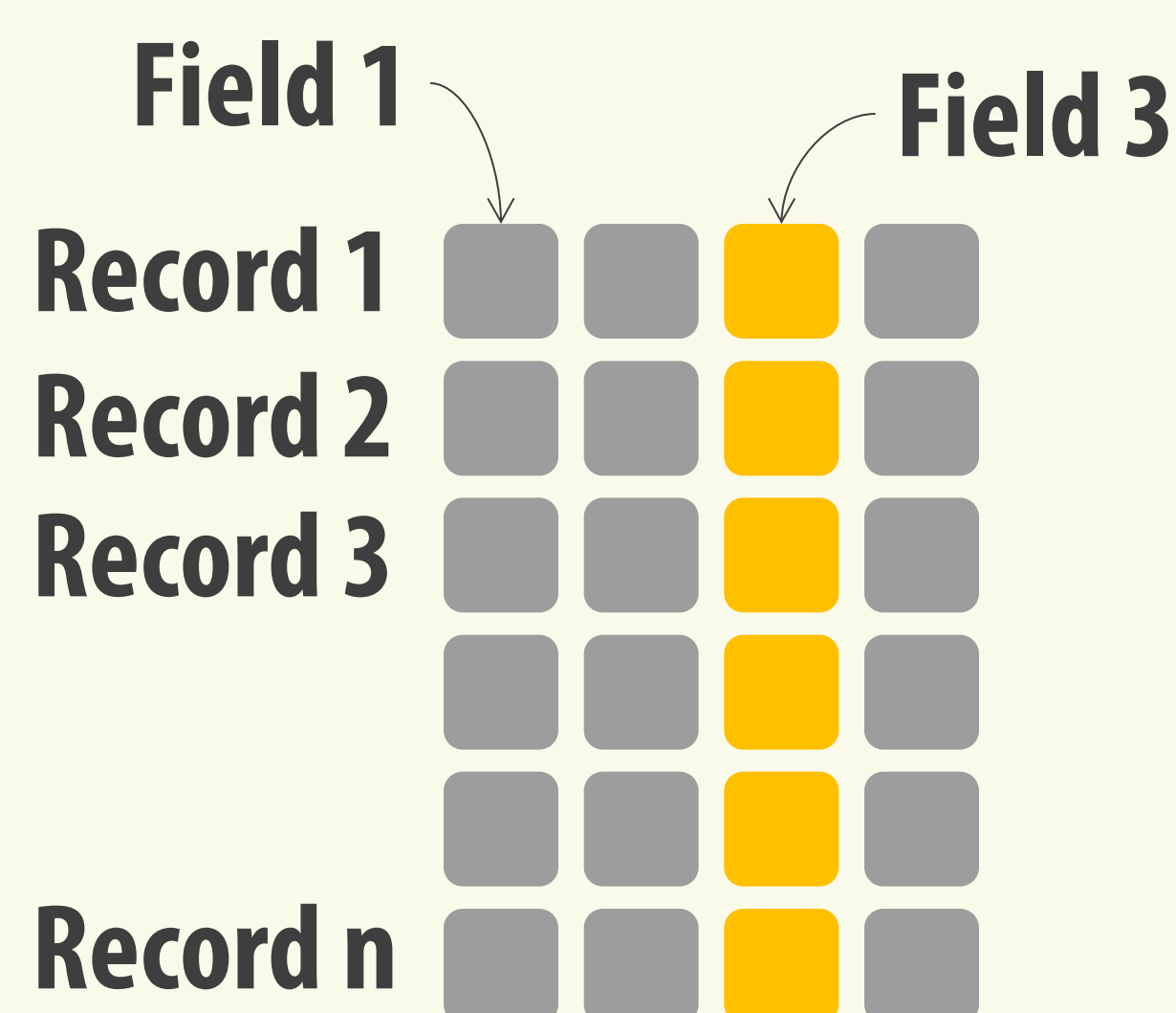


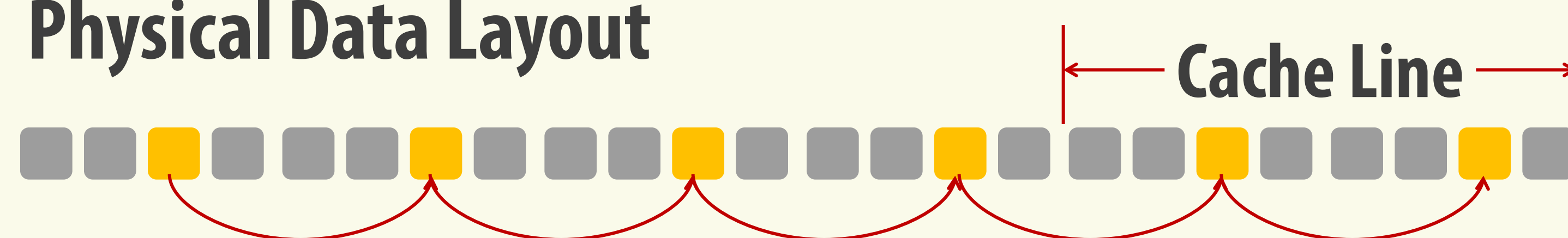
Gather-Scatter DRAM: Improving the Spatial Locality of Strided Access Patterns

Vivek Seshadri, Thomas Mullins, Amirali Boroumand, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry
Carnegie Mellon University, *Intel Labs

In-Memory Database Table

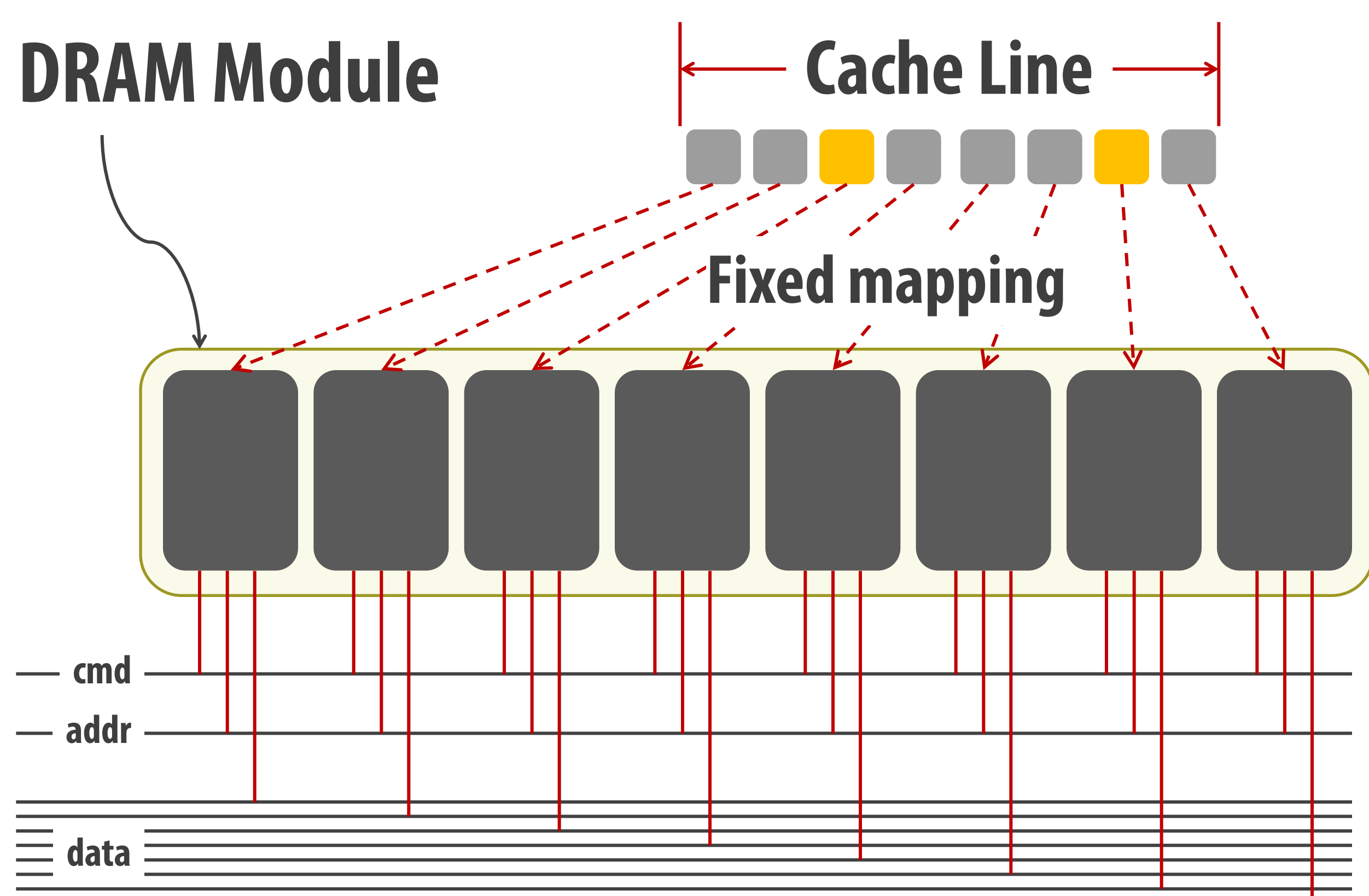


Physical Data Layout



1. High Latency
2. High Bandwidth Consumption
3. High Energy Consumption

DRAM Module

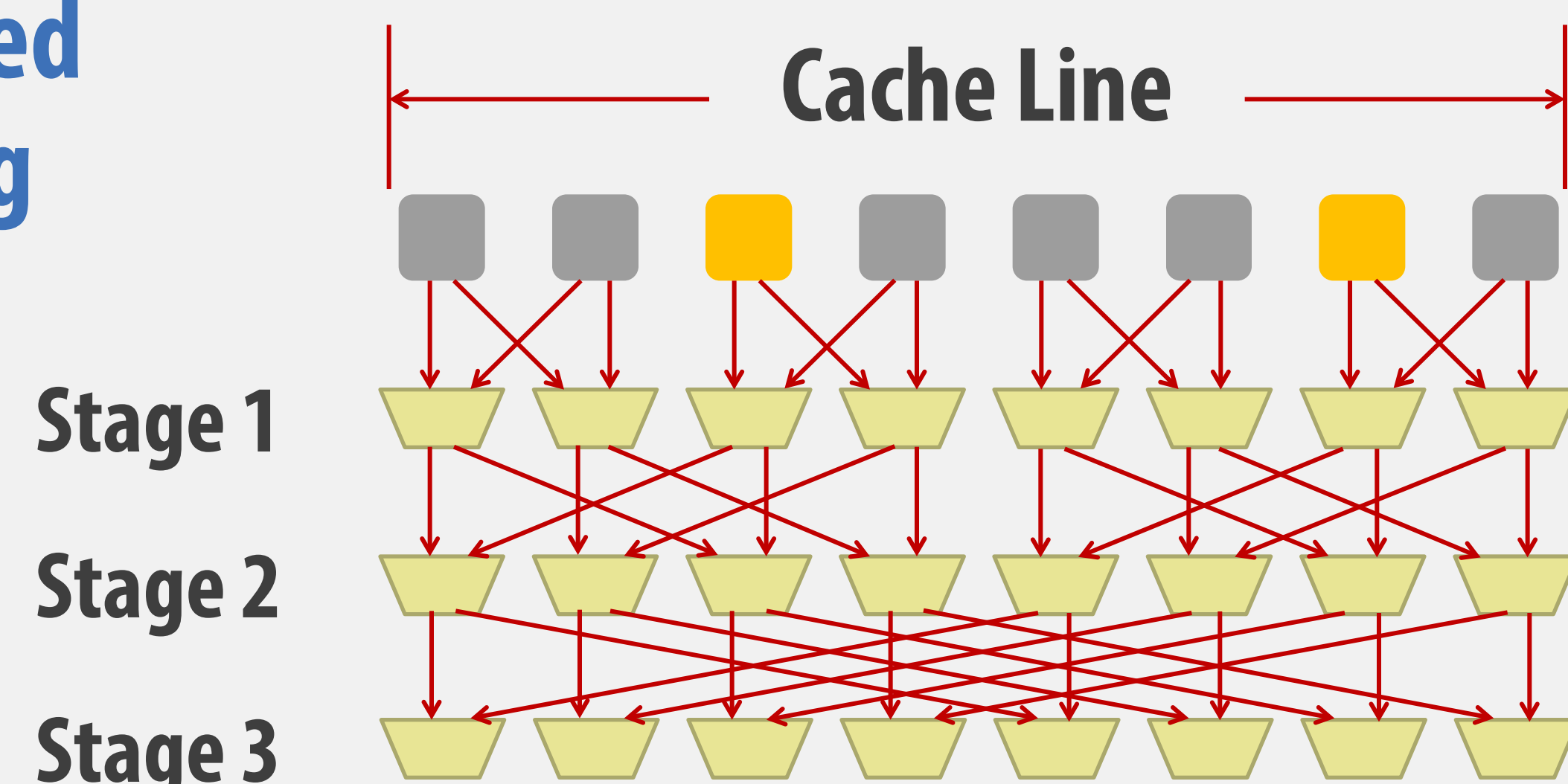


Observation: Each row buffer has many useful values

Idea: Gather a cache line of useful values in one read

Column ID-based Data Shuffling

Goal: Minimize chip conflicts for common patterns

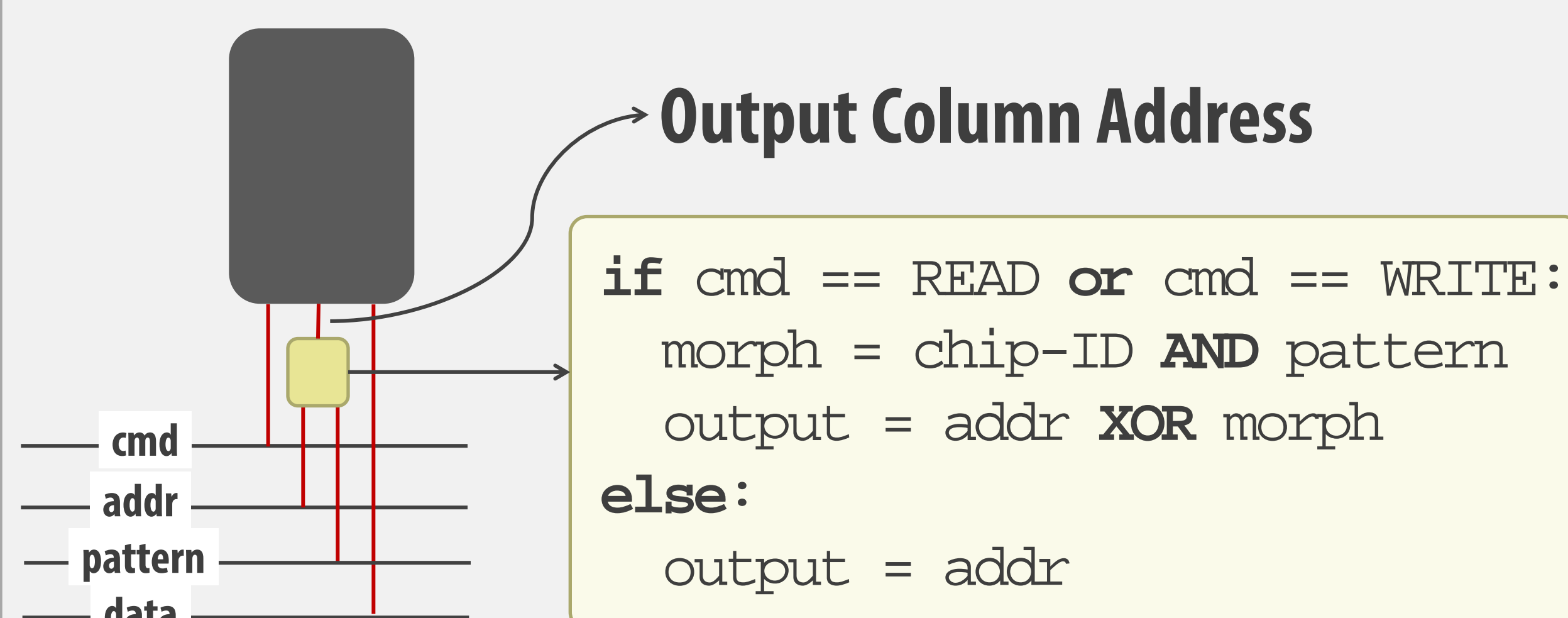


(stage 'n' is enabled only if nth least significant bit of column ID is set)

Gather/scatter many access patterns (e.g., any 2ⁿ stride) with near-ideal efficiency and latency!

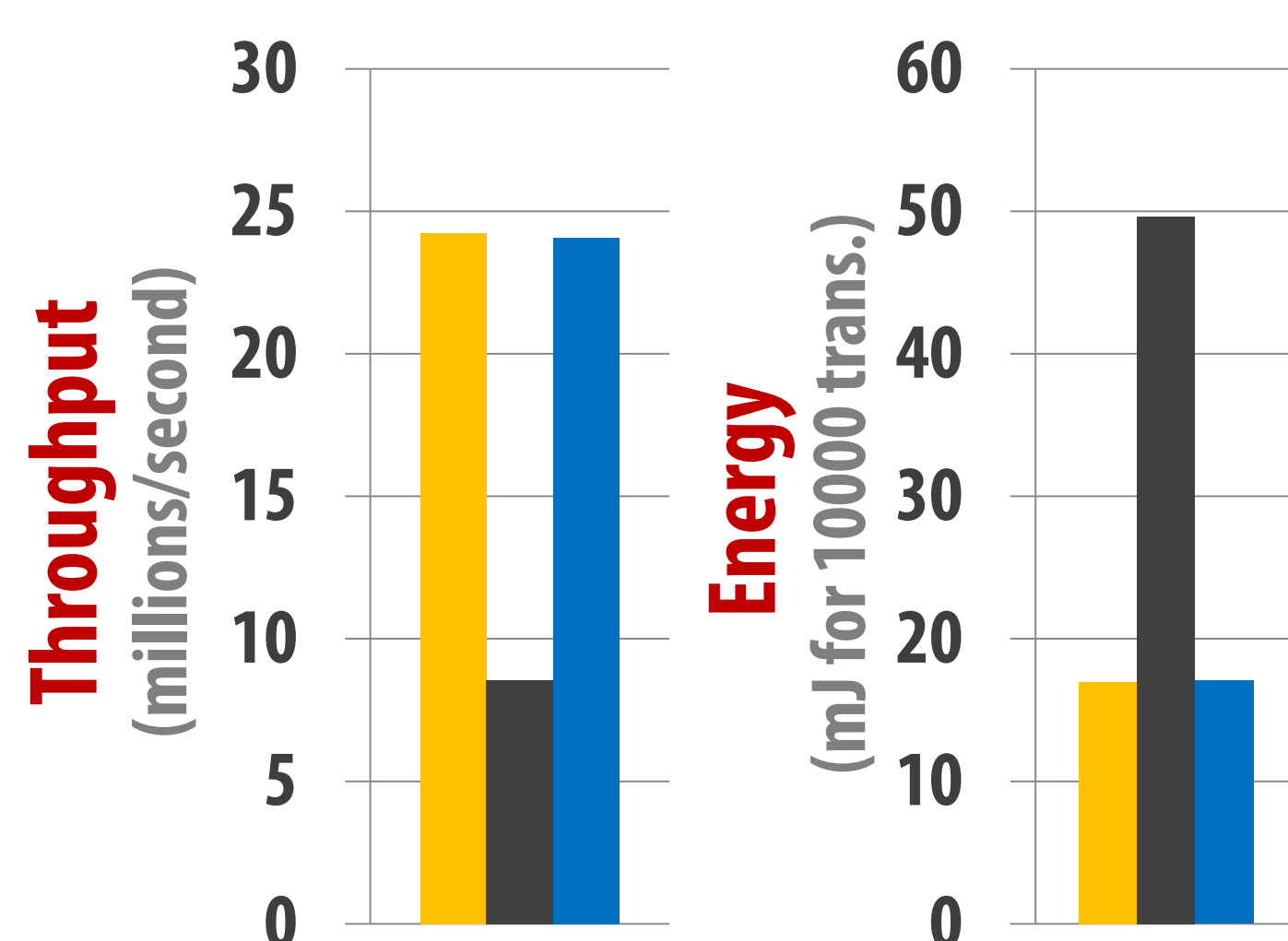
Minimal support from 1) on-chip caches, 2) instruction set architecture, and 3) software

Per-chip Column Translation Logic



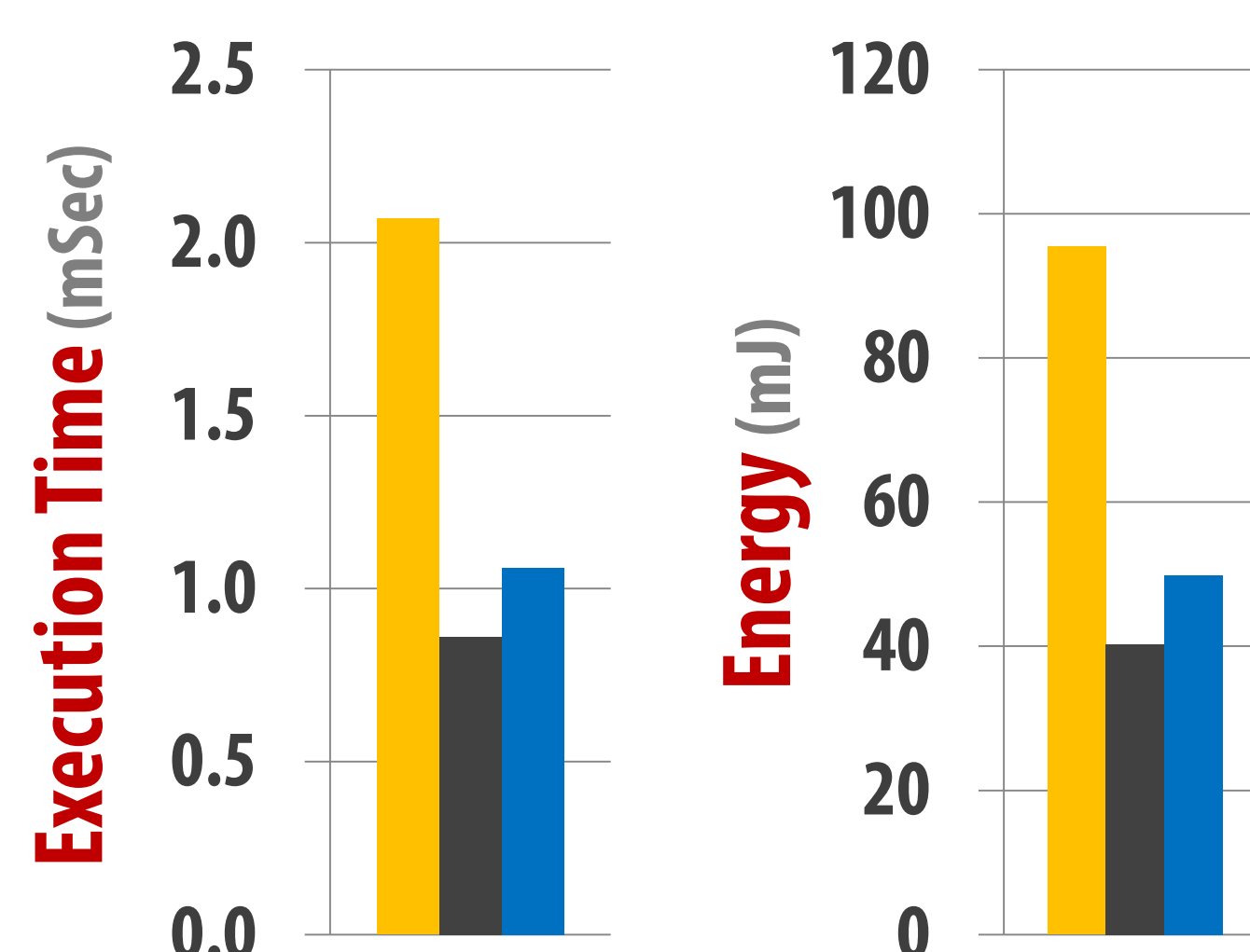
Transactions

(average across many workloads)
(varying number of R/W/RW fields)



Analytics

(average of two workloads)
(sum of 1 or 2 columns)



Hybrid Transactions/Analytics

(transactions = 1R+1W per record)
(analytics = sum of 1 column)

