Lecture 14: DRAM Main Memory Systems

- Today: cache/TLB wrap-up, DRAM basics (Section 2.3)

Superpages

- If a program’s working set size is 16 MB and page size is 8KB, there are 2K frequently accessed pages – a 128-entry TLB will not suffice

- By increasing page size to 128KB, TLB misses will be eliminated – disadvantage: memory waste, increase in page fault penalty

- Can we change page size at run-time?

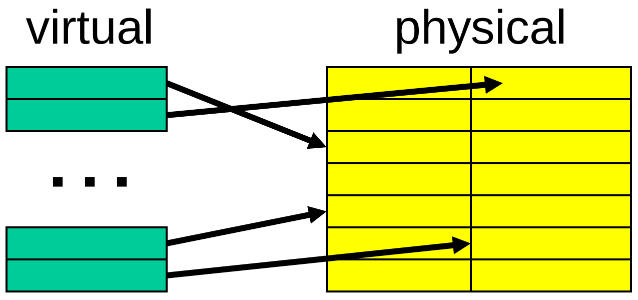
- Note that a single page has to be contiguous in physical memory

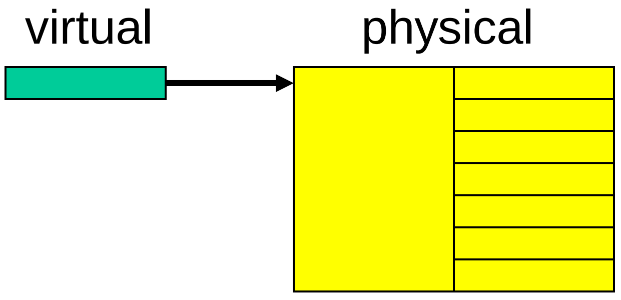
Superpages Implementation

- At run-time, build superpages if you find that contiguous virtual pages are being accessed at the same time

- For example, virtual pages 64-79 may be frequently accessed – coalesce these pages into a single superpage of size 128KB that has a single entry in the TLB

- The physical superpage has to be in contiguous physical memory – the 16 physical pages have to be moved so they are contiguous





Ski Rental Problem

- Promoting a series of contiguous virtual pages into a superpage reduces TLB misses, but has a cost: copying physical memory into contiguous locations

- Page usage statistics can determine if pages are good candidates for superpage promotion, but if cost of a TLB miss is *x* and cost of copying pages is *Nx,* when do you decide to form a superpage?

- If ski rentals cost $50 and new skis cost $500, when do I decide to buy new skis?
    - If I rent 10 times and then buy skis, I'm guaranteed to not spend more than twice the optimal amount

Prefetching

- Hardware prefetching can be employed for any of the cache levels

- It can introduce cache pollution – prefetched data is often placed in a separate prefetch buffer to avoid pollution – this buffer must be looked up in parallel with the cache access

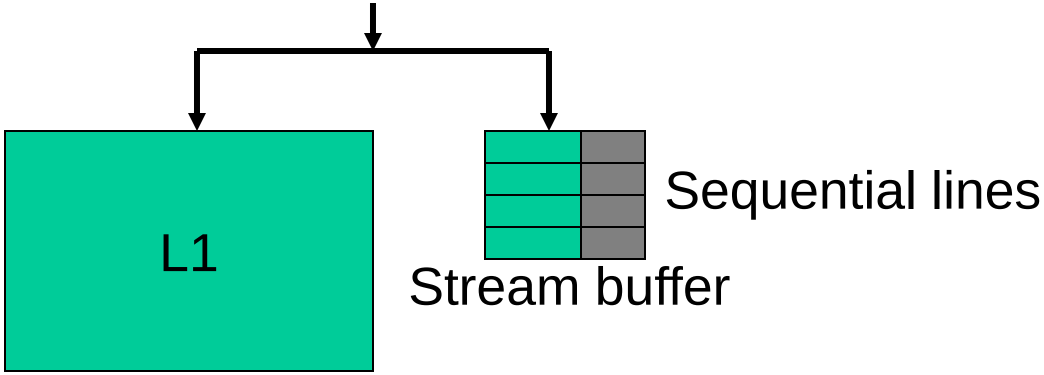
- Aggressive prefetching increases “coverage”, but leads to a reduction in “accuracy” → wasted memory bandwidth

- Prefetches must be timely: they must be issued sufficiently in advance to hide the latency, but not too early (to avoid pollution and eviction before use)

Stream Buffers

- Simplest form of prefetch: on every miss, bring in multiple cache lines

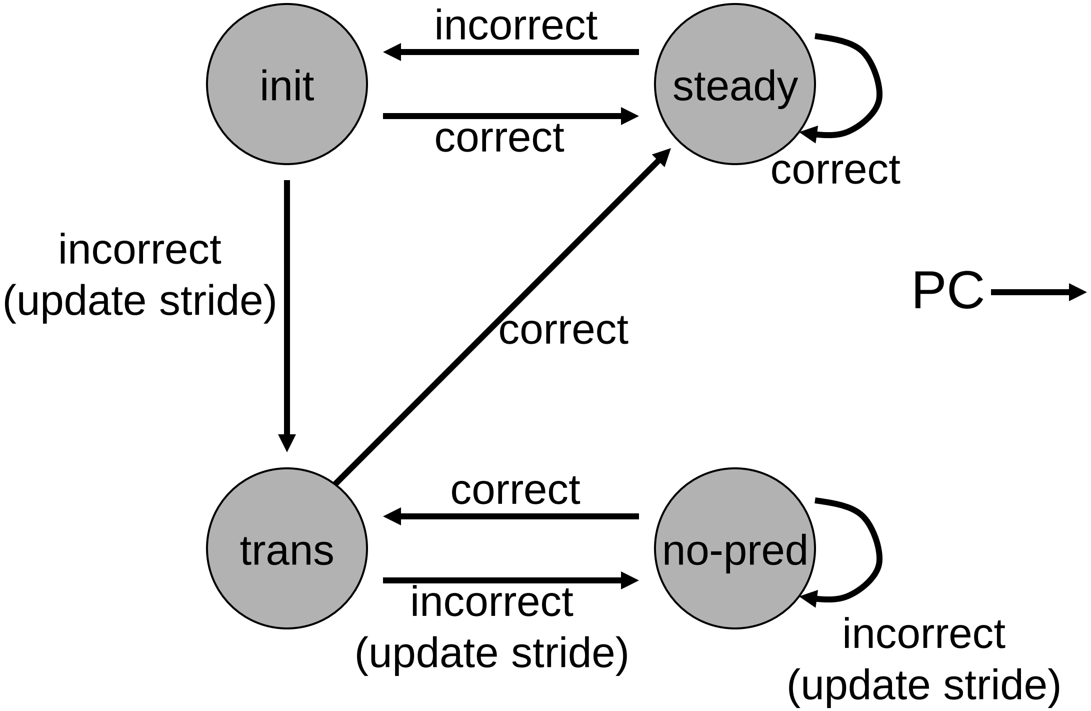
- When you read the top of the queue, bring in the next line



Stride-Based Prefetching

- For each load, keep track of the last address accessed by the load and a possibly consistent stride

- FSM detects consistent stride and issues prefetches



PC →

tag	prev_addr	stride	state

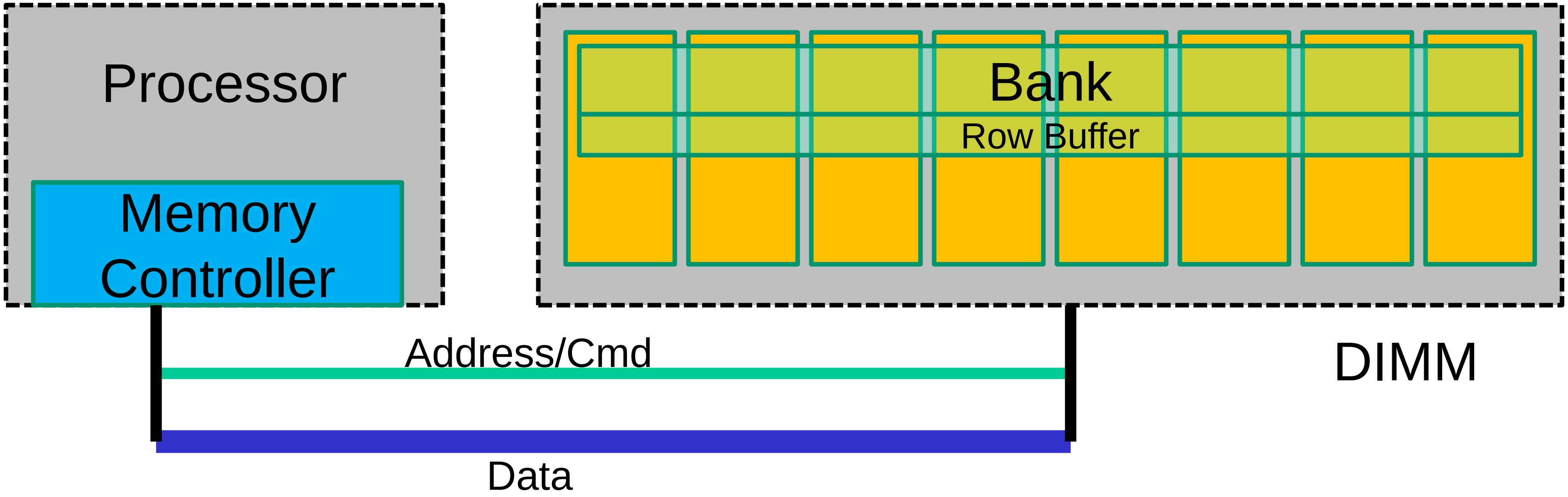
DRAM Main Memory

- Main memory is stored in DRAM cells that have much higher storage density

- DRAM cells lose their state over time – must be refreshed periodically, hence the name *Dynamic*

- DRAM access suffers from long access time and high energy overhead

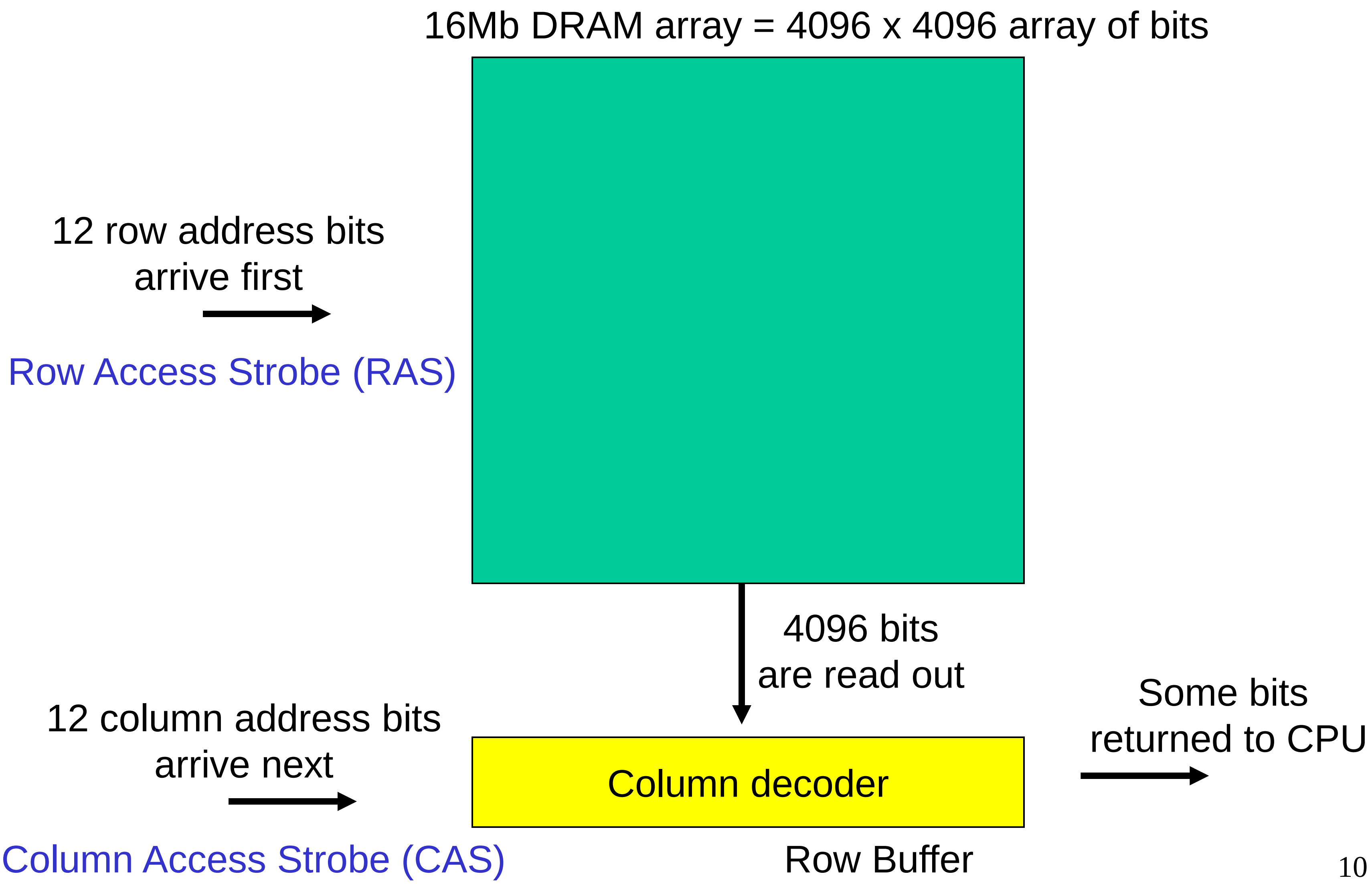
Memory Architecture



- DIMM: a PCB with DRAM chips on the back and front
- Rank: a collection of DRAM chips that work together to respond to a request and keep the data bus full
- A 64-bit data bus will need 8 x8 DRAM chips or 4 x16 DRAM chips or..
- Bank: a subset of a rank that is busy during one request
- Row buffer: the last row (say, 8 KB) read from a bank, acts like a cache

UTAH ARCH

DRAM Array Access



16Mb DRAM array = 4096 x 4096 array of bits

12 row address bits
arrive first

Row Access Strobe (RAS)

4096 bits
are read out

12 column address bits
arrive next

Column Access Strobe (CAS)

Column decoder

Row Buffer

Some bits
returned to CPU

Organizing a Rank

- DIMM, rank, bank, array → form a hierarchy in the storage organization
- Because of electrical constraints, only a few DIMMs can be attached to a bus
- One DIMM can have 1-4 ranks
- For energy efficiency, use wide-output DRAM chips – better to activate only 4 x16 chips per request than 16 x4 chips
- For high capacity, use narrow-output DRAM chips – since the ranks on a channel are limited, capacity per rank is boosted by having 16 x4 2Gb chips than 4 x16 2Gb chips

Organizing Banks and Arrays

- A rank is split into many banks (4-16) to boost parallelism within a rank
- Ranks and banks offer memory-level parallelism
- A bank is made up of multiple arrays (subarrays, tiles, mats)
- To maximize density, arrays within a bank are made large → rows are wide → row buffers are wide (8KB read for a 64B request, called overfetch)
- Each array provides a single bit to the output pin in a cycle (for high density)

Row Buffers

- Each bank has a single row buffer

- Row buffers act as a cache within DRAM
  - Row buffer hit: ~20 ns access time (must only move data from row buffer to pins)
  - Empty row buffer access: ~40 ns (must first read arrays, then move data from row buffer to pins)
  - Row buffer conflict: ~60 ns (must first precharge the bitlines, then read new row, then move data to pins)

- In addition, must wait in the queue (tens of nano-seconds) and incur address/cmd/data transfer delays (~10 ns)

Reads and Writes

- A single bus is used for reads and writes
- The bus direction must be reversed when switching between reads and writes; this takes time and leads to bus idling
- Hence, writes are performed in bursts; a write buffer stores pending writes until a high water mark is reached
- Writes are drained until a low water mark is reached

Open/Closed Page Policies

- If an access stream has locality, a row buffer is kept open
  - Row buffer hits are cheap (open-page policy)
  - Row buffer miss is a bank conflict and expensive because precharge is on the critical path

- If an access stream has little locality, bitlines are precharged immediately after access (close-page policy)
  - Nearly every access is a row buffer miss
  - The precharge is usually not on the critical path

- Modern memory controller policies lie somewhere between these two extremes (usually proprietary)

Address Mapping Policies

- Consecutive cache lines can be placed in the same row to boost row buffer hit rates
- Consecutive cache lines can be placed in different ranks to boost parallelism
- Example address mapping policies:

  row:rank:bank:channel:column:blkoffset

  row:column:rank:bank:channel:blkoffset

Scheduling Policies

- FCFS: Issue the first read or write in the queue that is ready for issue

- First Ready - FCFS: First issue row buffer hits if you can

- Stall Time Fair: First issue row buffer hits, unless other threads are being neglected

Refresh

- Every DRAM cell must be refreshed within a 64 ms window
- A row read/write automatically refreshes the row
- Every refresh command performs refresh on a number of rows, the memory system is unavailable during that time
- A refresh command is issued by the memory controller once every 7.8us on average

Title

- Bullet