Having 128GB of

Storage



Ram



V Ram



L cache



Lab 3 graded

- Find grade on krebs

TODO

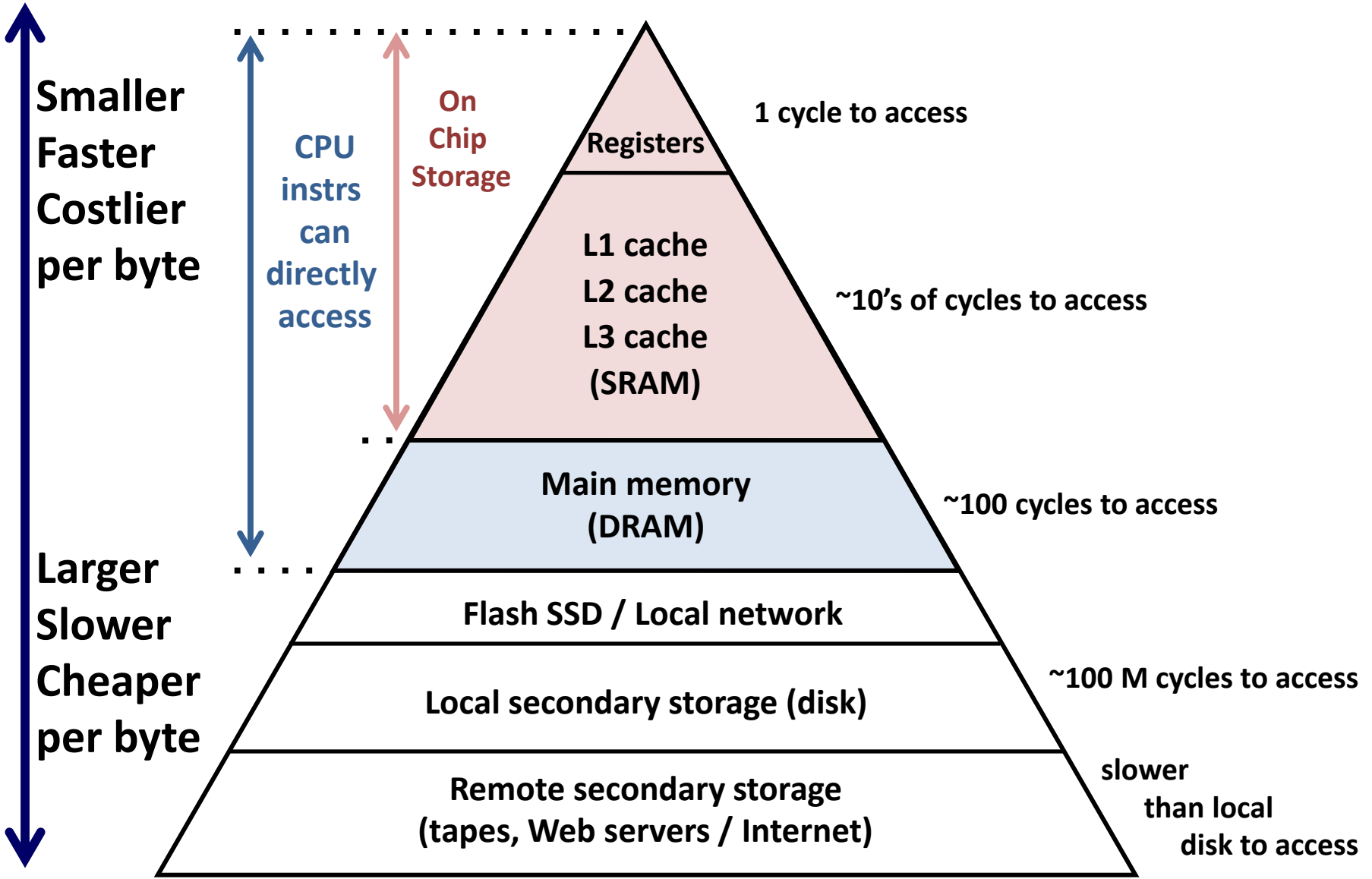
- Assignment 4 assigned due November 10
- Study for Exam 2 November 6

Overview

- Conclusion of memory hierarchy

Caches

- Caches are used to bridge performance gaps between smaller, faster layers and larger, slower layers


Smaller
Faster
Costlier
per byte
Larger
Slower
Cheaper
per byte
CPU instrs can directly access
On Chip Storage
Registers
1 cycle to access
L1 cache
L2 cache
L3 cache
(SRAM)
~10's of cycles to access
Main memory
(DRAM)
~100 cycles to access
Flash SSD / Local network
Local secondary storage (disk)
~100 M cycles to access
Remote secondary storage
(tapes, Web servers / Internet)
slower
than local
disk to access

Caches

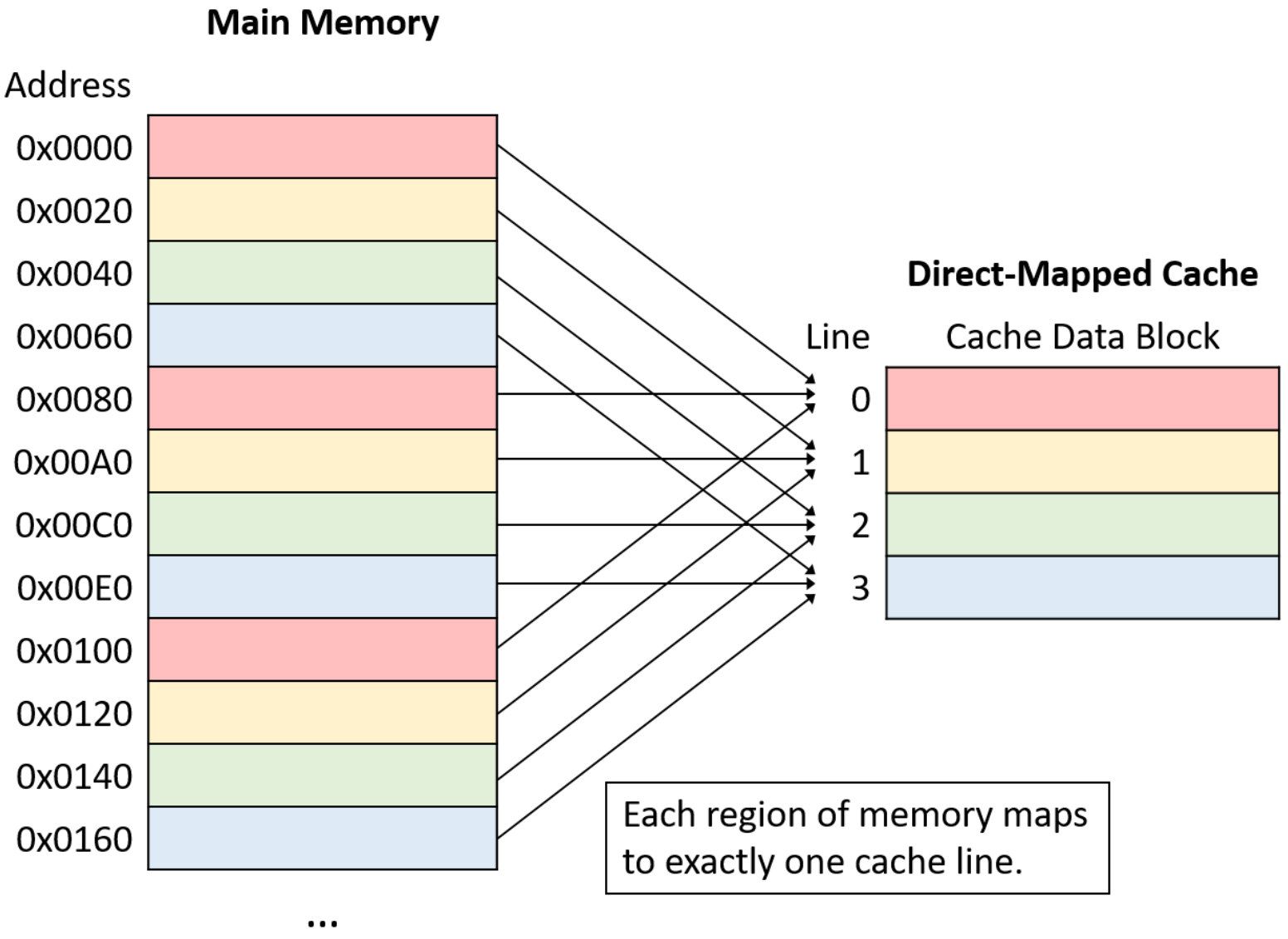
- Caches are used to bridge performance gaps between smaller, faster layers and larger, slower layers
- CPU caches bridge between registers and main memory

Where to look in the cache

- I want what's at address 70.
  - Where should I look for it in the cache?
- First determine the block # *n*:
  - $n = \lfloor 70 / 64 \rfloor$ (assuming a 64 byte block)
- Where is block *n* in the cache?

Direct mapping

- Direct mapping is simplest block placement
- Each block maps to exactly one cache line


Main Memory
Address
0x0000
0x0020
0x0040
0x0060
0x0080
0x00A0
0x00C0
0x00E0
0x0100
0x0120
0x0140
0x0160
...
Direct-Mapped Cache
Line
Cache Data Block
0
1
2
3
Each region of memory maps to exactly one cache line.

Selecting cache line

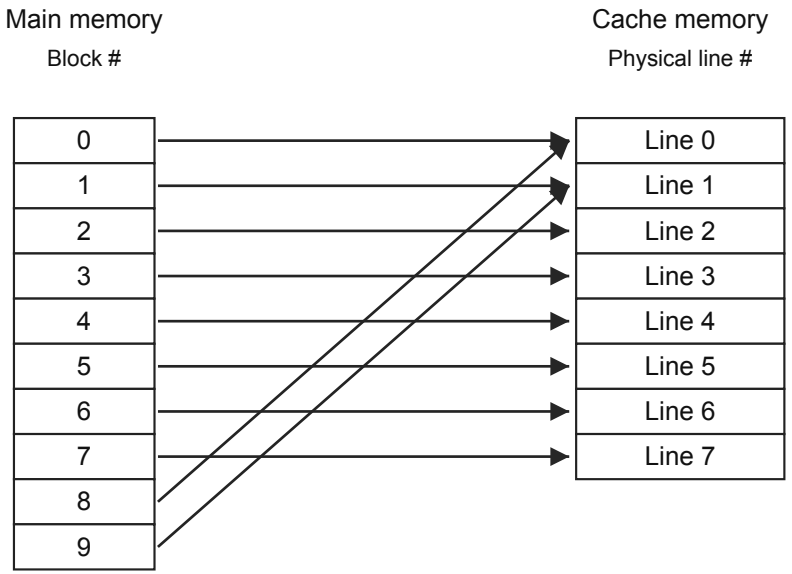
- For memory block numbered n,
  select cache line $\ell = n \% L$, where L is the total number of cache lines

Reduce placement conflicts

- Allow a block to be in more than one cache line
- Set-associative cache – each block maps to a set of cache lines
- 2-way associative – a block maps to a set of 2 cache lines
- *n*-way associative – a block maps to a set of *n* cache lines…

Direct-mapped cache

S = 8 sets • E = 1 line/set • C = 8B

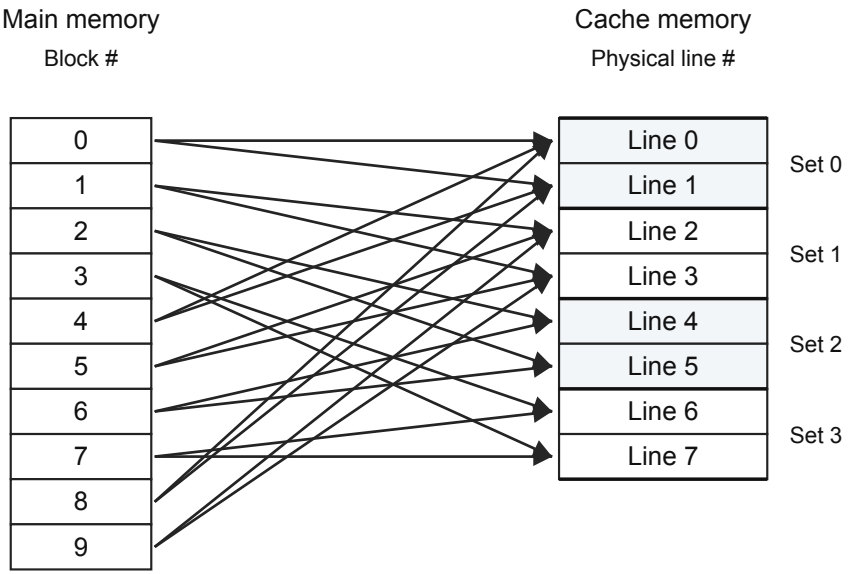


Each block maps to exactly one physical line.

Line number $\ell = n \bmod 8$.

2-way set-associative cache

S = 4 sets • E = 2 lines/set • C = 8B



Each block can occupy either of two lines in its set.

For set $s = n \bmod 4$, use lines $2s$ and $2s + 1$.

S/E/B cache model

- A cache can be described by three values
  - *S*: number of cache sets
  - *E*: number of lines per set a.k.a. the associativity
  - *B*: number of bytes per block
- The cache size $C = S * E * B$

S/E/B cache model

- A cache can be described by three values
  - *S*: number of cache sets
  - *E*: number of lines per set a.k.a. the associativity
  - *B*: number of bytes per block
- For memory block numbered *n*,
  select set $s = n \% S$
  - The first cache line of *s* is $\ell = s * E$
  - The last cache line of *s* is $\ell' = s * (E + 1) - 1$

Cache miss → Which to evict?

- Direct-mapped: only one choice!
- *n*-way set associative: *n* choices!
- Which should we evict?

Cache miss → Which to evict?

- Direct-mapped: only one choice!
- *n*-way set associative: *n* choices!
- Which should we evict?
  - LRU cache – evict least recently used
  - Modern caches use proprietary algorithms but have LRU-like eviction

- Consider a direct-mapped cache with eight lines and 1-byte blocks.
- For each accessed address below, record whether the access is a H (hit) or M (miss).
- For each miss, record the block evicted if a block was evicted.
  - 5, 7, 6, 2, 1, 4, 3, 0,
    5, 7, 6, 2, 1, 4, 3, 0,
    5, 7, 6, 2, 1, 4, 3, 0

- Consider a direct-mapped cache with eight lines and 1-byte blocks.
- For each accessed address below, record whether the access is a H (hit) or M (miss).
- For each miss, record the block evicted if a block was evicted.
  - 8, 3, 6, 2, 7, 5, 1, 9, 0, 4,
    8, 3, 6, 2, 7, 5, 1, 9, 0, 4,
    8, 3, 6, 2, 7, 5, 1, 9, 0, 4

Observation #1

- When the # of blocks in this cycle exceeds the # of cache lines, we start seeing misses!

- Consider a four-way set-associative cache (i.e., **E** = 4 lines per set) with **S** = 2 sets, **B** = 1-byte blocks, and LRU replacement with total cache size **C** = 8 bytes.
- For each accessed address below, record whether the access is a H (hit) or M (miss).
- For each miss, record the block evicted if a block was evicted.
  - 24, 0, 16, 8,
    24, 0, 16, 8,
    24, 0, 16, 8

- Consider a four-way set-associative cache (i.e., **E** = 4 lines per set) with **S** = 2 sets, **B** = 1-byte blocks, and LRU replacement with total cache size **C** = 8 bytes.
- For each accessed address below, record whether the access is a H (hit) or M (miss).
- For each miss, record the block evicted if a block was evicted.
  - 24, 16, 32, 8, 0,
    24, 16, 32, 8, 0,
    24, 16, 32, 8, 0

Observation #2

- When the # of blocks in this cycle exceeds the associativity, we start seeing misses!
- Note: unlike in the previous exercise, in this one we are trying to always hit the same cache set

How to shuffle an array?

- Let’s see…

How to shuffle an array?

- Fisher-Yates shuffle is correct!
- YOLO-shuffle is probably not!

On a future lab

- You will be using these observations and Fisher-Yates shuffle to measure real-world cache parameters
  - Cache sizes (L1, L2, L3)
  - Associativity (L1)

Structure of course

- First third (Exam 1):
  - Data Representation
  - Memory
- Second third (Exam 2):
  - Machine Code
  - Memory exploits
- Final third (Final exam):
  - Memory hierarchy
  - Processes ← here

Work on Lab 4

- All done with caches!
- No in-class exercise today
- Spend rest of period working on Lab 4

TODO

- Assignment 4 assigned due November 10
- Study for Exam 2 November 6