Lecture 16: Non-blocking caches

Thursday, March 1, 2018 9:40 AM

Outline

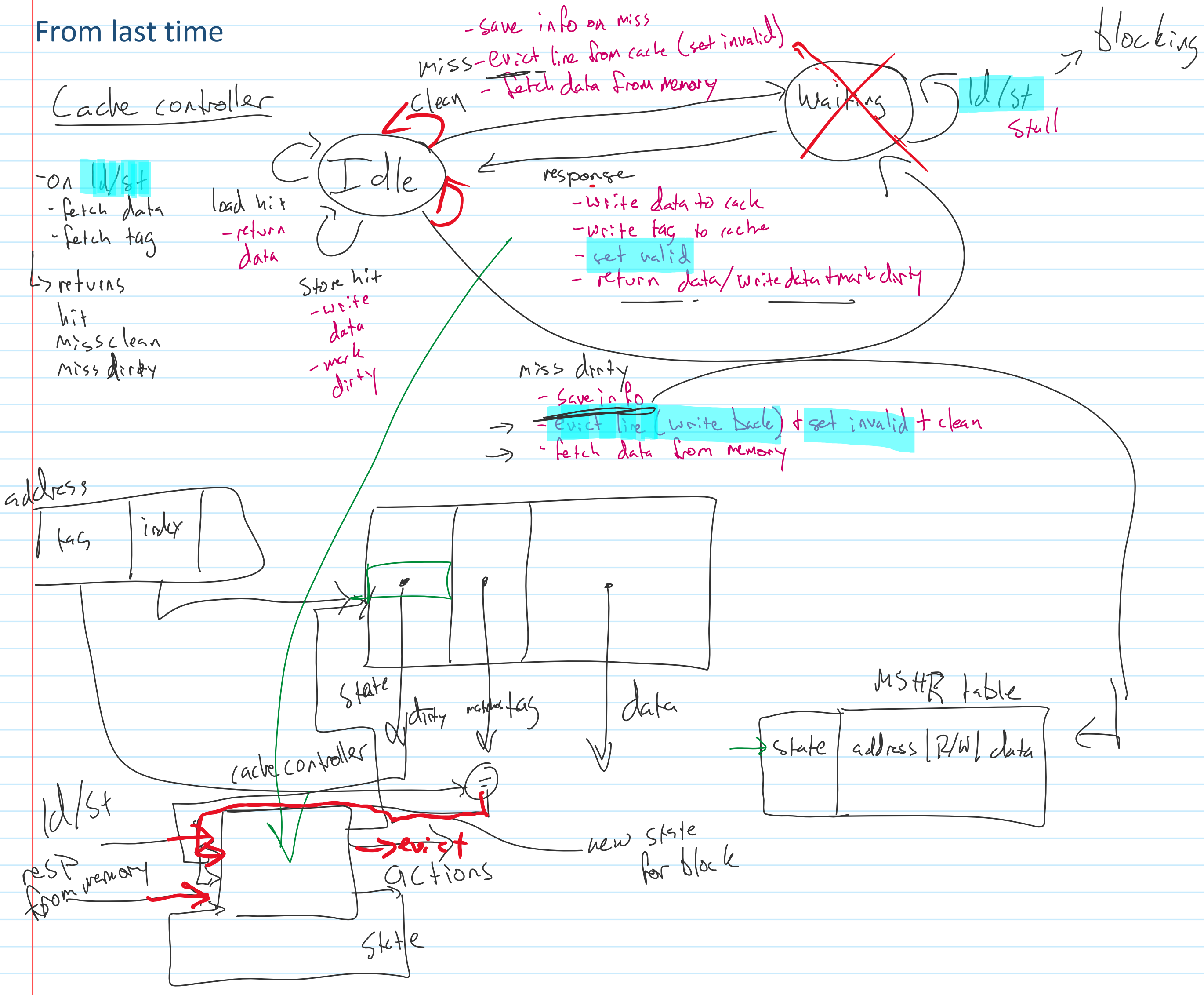
- Talk about non-blocking cache controller design
- Talk about Lab 4
- Replacement policies for set associative caches

From last time

Cache controller

- on ld/st
  - fetch data
  - fetch tag

↳ returns
hit
miss clean
miss dirty

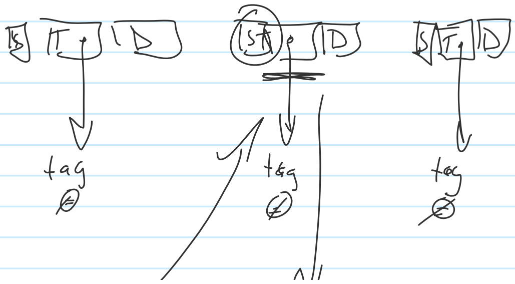


For non blocking cache: Still need to track states

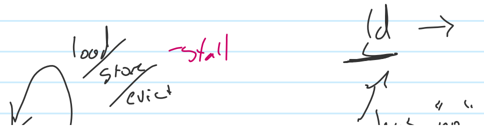
- now tracking state at line/block granularity
- a state machine for every block

→ a way to track outstanding requests

↳ miss status handling register (MSHR)



Block-level state machine for a non-blocking



Block-level State machine
for a non-blocking
write back cache

Possible States?

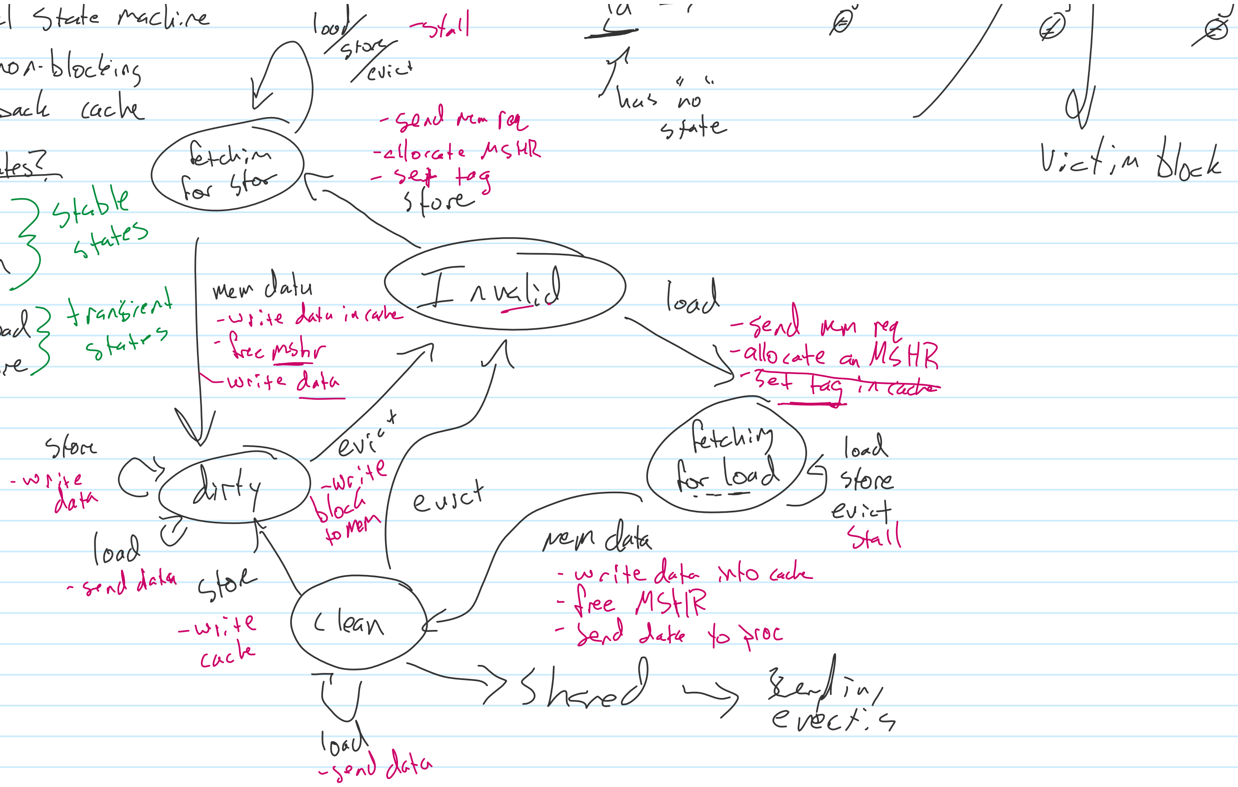
- Invalid
- ~~Valid~~ Clean
- Dirty
- fetching load
- fetching store

stable states

transient states

Events

- load
- Store
- mem data
- evict



Replacement policies

How to choose what to evict?

Least recently used (LRU) evict the item used furthest in past
+ temporal locality
+ close to theoretical best

- expensive in bits, and hardware
- every access is expensive

round robin way 0, way 1 ...

Clock algorithm -> give things a "time"

Random

not most recently used NMRU ←

Vary w/ level of cache