Essential requirements for a coherence protocol?

- Enforce an ordering on key messages about the same block
- Ideally is architecturally invisible

How do you build an efficient snooping protocol?

- Need a broadcast bus

- Which states are supported matter
  - Owner state is an optimization of MESI
- Construct a *logical* bus atop a network
- Split busses along functionality

How do you make directory coherence efficient?

- At the memory system
  - In last-level-cache: if a cache line isn't in *any* cache no need for a directory entry
    - Scale via hierarchy
    - Scale via partitioning
- Handling >> readers
  - LRU among finite number of sharers
  - Everybody state
  - group cores into shared groups (hierarchy helps here)
  - slow fall-back (linked list of readers?)

What is token coherence?

- All tokens = can read or write
- Any token = can read
-

Do you really need coherence? What if you didn't have HW coherence, then what?

- Resolve multiple modifications to the same cache line by merging
- Minimize (eliminate?) sharing, prohibit process migration
- Take a message passing approach
    - Treat cache as program-managed structure
- Assume data-race-free code, use explicit cache flushes as needed
- Assume data-race free code, flush on lock-acquire, and have write-through caches