CS305
Computer Architecture

Cache Performance Analysis

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

Outline

- The three kinds of cache misses
- Performance implications of block size
- Joint I+D cache vs separate I, D caches
- Performance implications of associativity
- Design of cache ←→ main memory interface
- Multi-level caches

The Three Kinds of Cache Misses

Compulsory	The miss caused the first time a block is accessed Since the cache starts “cold”, “empty”…
Conflict	A miss caused due to insufficient set size Cannot happen in a fully associative cache (by defn.)
Capacity	A miss caused due to insufficient cache size All misses after compulsory misses, in fully assoc. caches

Effect of Increase in Block Size

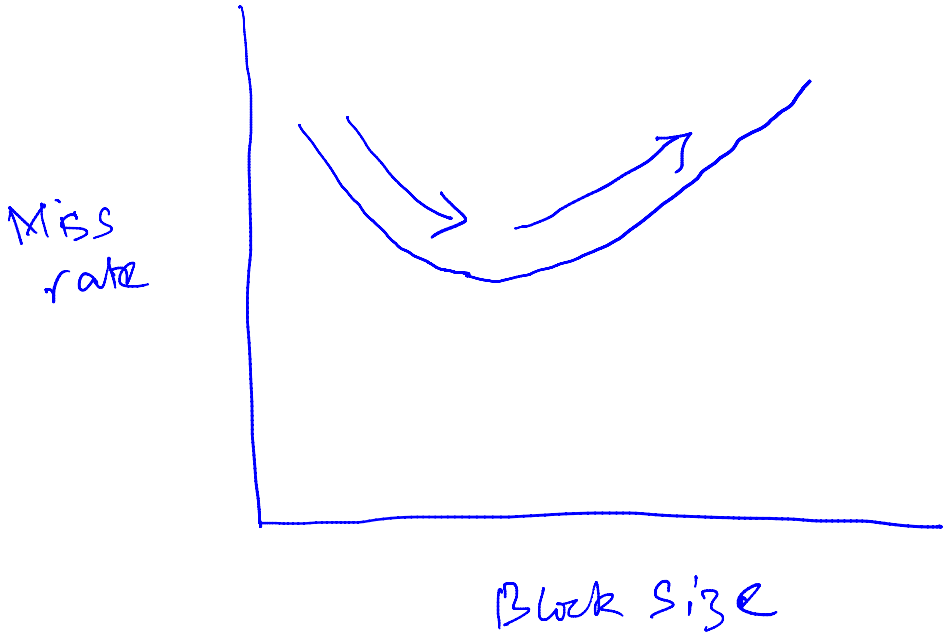
(+) Increased spatial locality

➔ Lesser compulsory misses

(–) More conflict misses: for same cache size

(–) Higher miss penalty

Miss Rate vs Block Size



Increased Block Size: Techniques to Reduce Miss Penalty

- Early restart
  - Processor can proceed as soon as required word is in cache
- Critical word first
  - Get the word required by the processor first, then the rest of the block
- Implications: increased complexity in cache controller and/or memory system

Joint I+D Cache or Separate I, D Caches

Joint I+D Cache

(+) Better hit rate

(–) Lower instruction throughput (pipeline stalls)

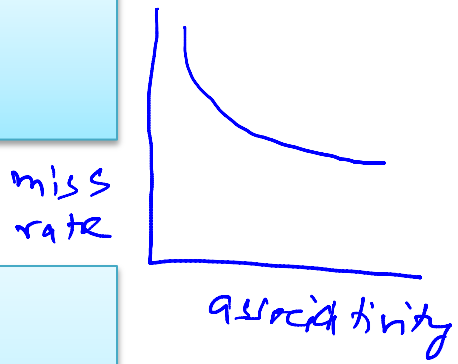
Separate I, D Caches

(–) Slightly lower hit rate

(+) Better instruction throughput

Performance Implications of Associativity

(+) Reduced conflict misses
(–) Increased hit time!



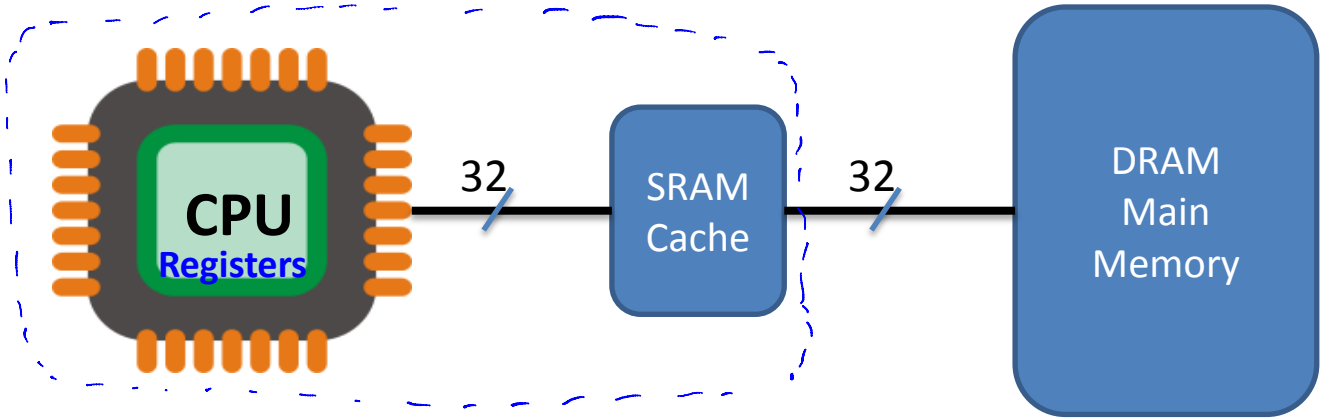
Note: decreasing benefits of higher associativity.
Reason: only a certain number of conflict misses to remove

Design of
Cache-to-Main-Memory Interface

What Happens on a Cache Miss?

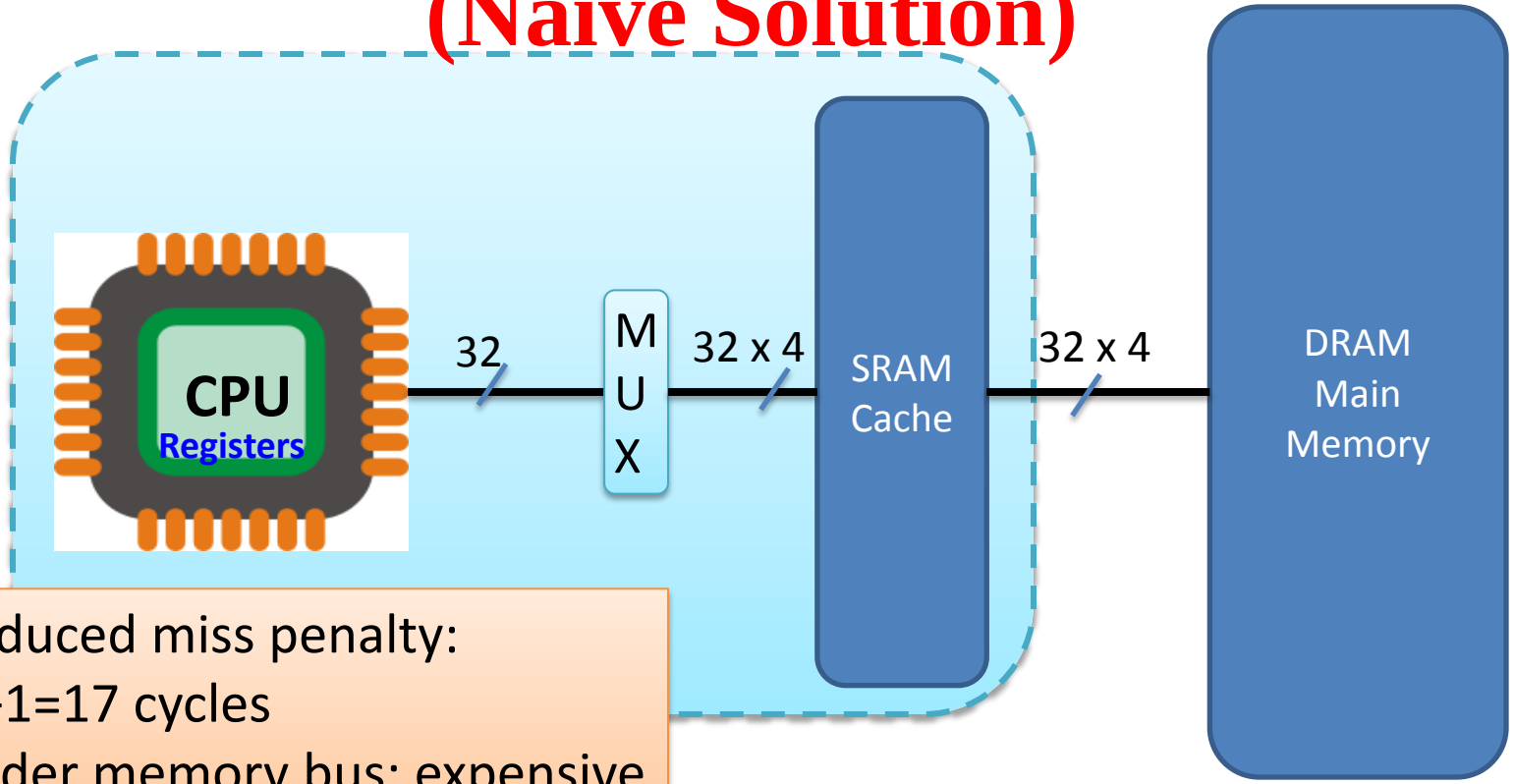
Miss penalty: time to load a block from main memory to cache

a) Send address to memory	→ Say 1 cycle
b) DRAM access initiation latency	→ Say 15 cycles
c) Read 1 word of data from memory to cache	→ Say 1 cycle



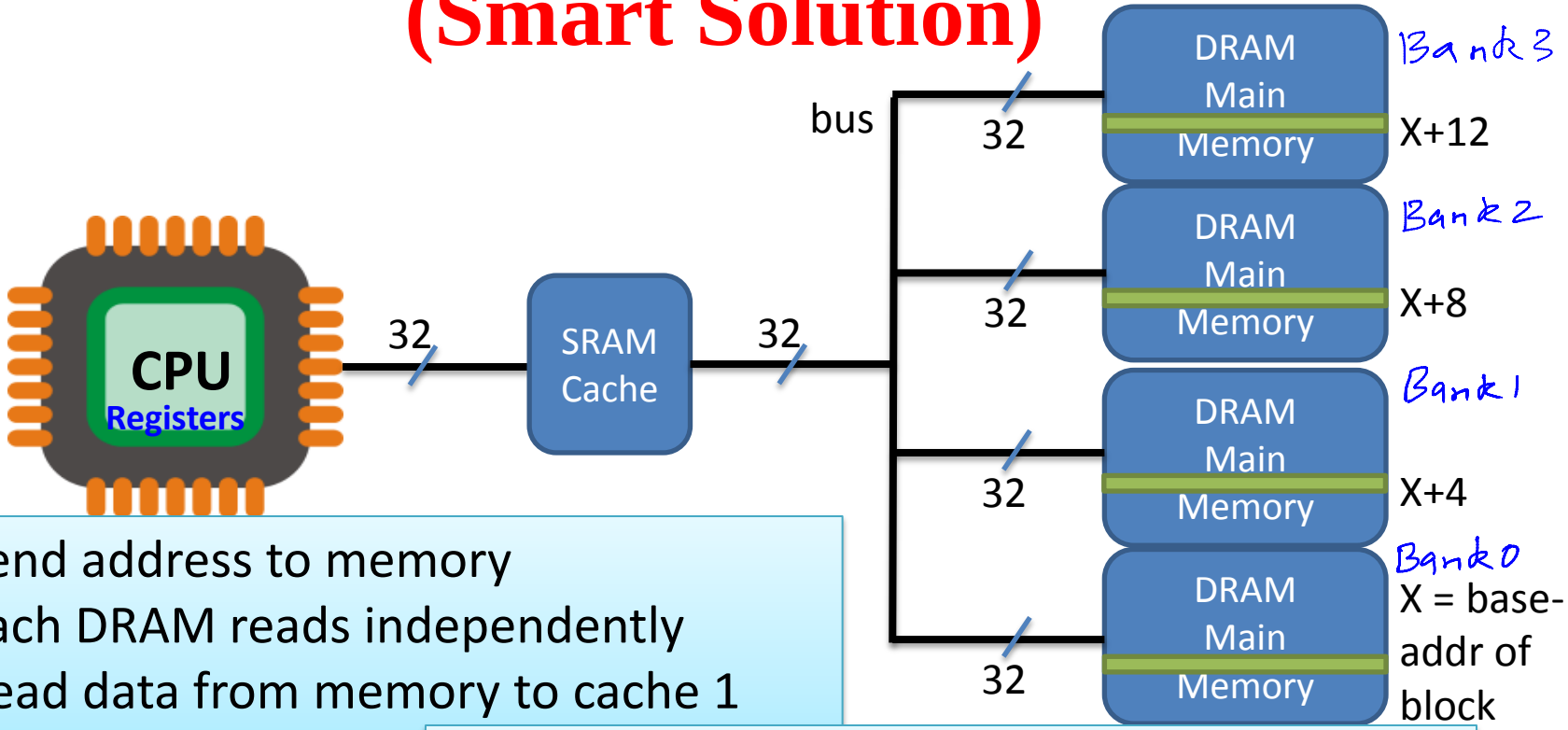
Miss penalty for 4-word block = 4 x (1+15+1) = 68 cycles

Reducing Miss Penalty: Wide Memory (Naïve Solution)



(+) Reduced miss penalty:
1+15+1=17 cycles
(–) Wider memory bus: expensive
(–) Increased hit time!

Reducing Miss Penalty: Interleaved Memory (Smart Solution)

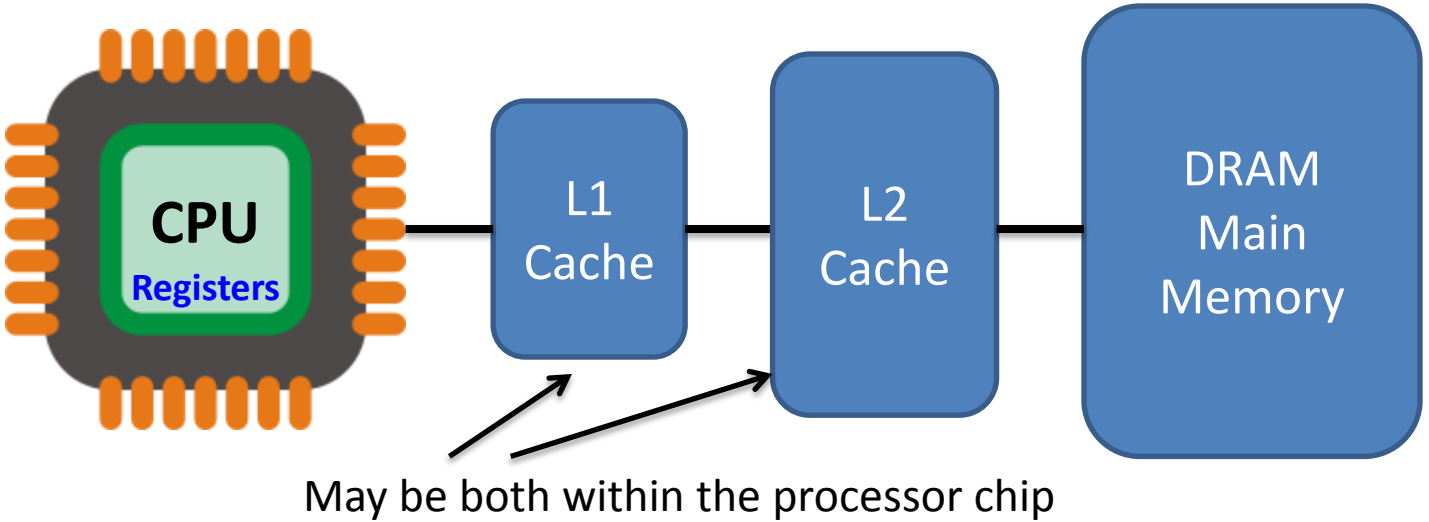


a) Send address to memory
b) Each DRAM reads independently
c) Read data from memory to cache 1 word after another

Miss penalty for 4-word block = 1+15+4x1 = 20 cycles

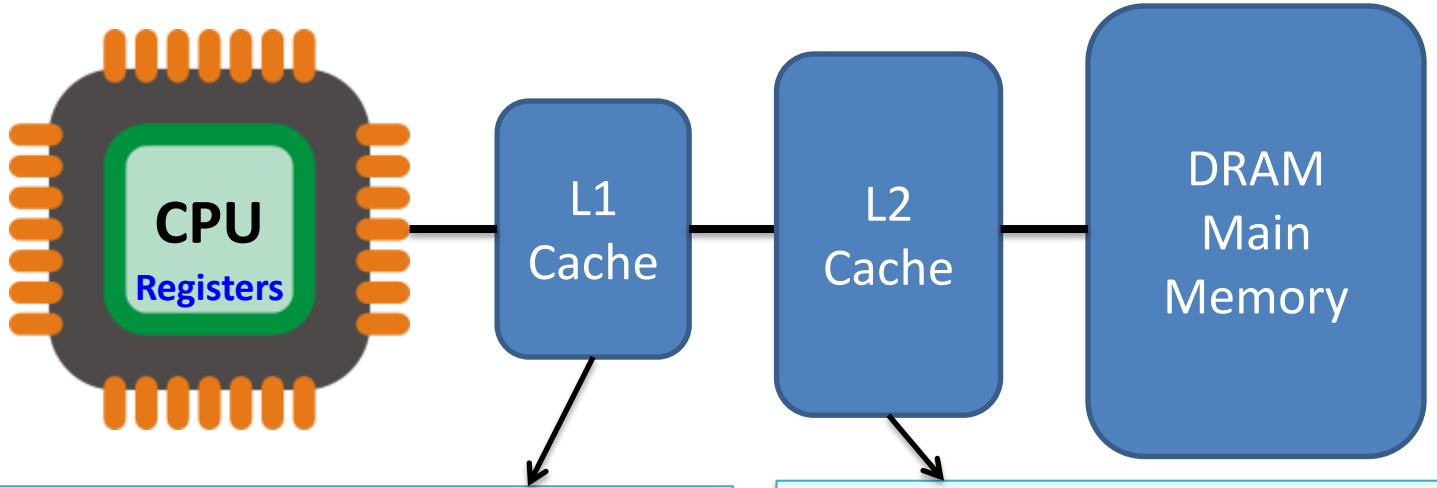
Multi-Level Caches

Reducing Miss Penalty: Multi-Level Caches



L1 thinks L2 to be main memory, L2 thinks L1 is processor
Miss in L1 ➔ see in L2, Miss in L2 ➔ see in main memory

Reducing Miss Penalty: Multi-Level Caches



Optimize for hit-time:
(miss penalty low anyway)
Smaller size, smaller blocks
Direct mapped or low associativity

Optimize for miss-rate:
(hit time does not matter anyway)
Larger size, larger blocks
2, 4, or 8-way associative

L2 Miss Rate: Local vs Global

L2 **local** miss rate: with respect to L2 accesses

L2 **global** miss rate: with respect to memory references by processor

Summary

- The three C’s: compulsory, conflict, capacity
- Performance implications of design options: block size, separate vs unified, associativity
- Interleaved memory
- Multi-level caches
- Next: program performance in presence of caches