CS 354 - Machine Organization & Programming
Tuesday, October 17, 2017

Project p3 (6%): DUE at 10 pm on Wednesday, November 1st

Homework hw4 (1.5%): Assigned tonight

Last Time

Explicit Free List
Ordering Free Blocks
Heap Caveats
Locality

Today

Locality (from last time)
Memory Hierarchy (from last time)
Cache Basic Idea
Designing a Cache - Blocks
Designing a Cache - Sets and Tags
Designing a Cache - Lines
Basic Cache Operation

Next Time

More Cache Design and Operation
Re-read: B&O 6.4.2

Cache Basic Idea

cache is a Smaller, faster storage device that acts as a staging area for data stored in a larger slower device

22,11,22,44,11,33,11,22,55,27,44

cache miss

When the block is not found in cache

cold miss

When the cache block is empty or invalid

capacity miss

Cache is too small for the working set

conflict miss

blocks map to the same cache set

cache hit

Found in cache

placement policy

- Anywhere
- Restrictive maps block to 1 location

replacement policy

- Level 1: Least recently used
- No choice, singleton
- Random choice

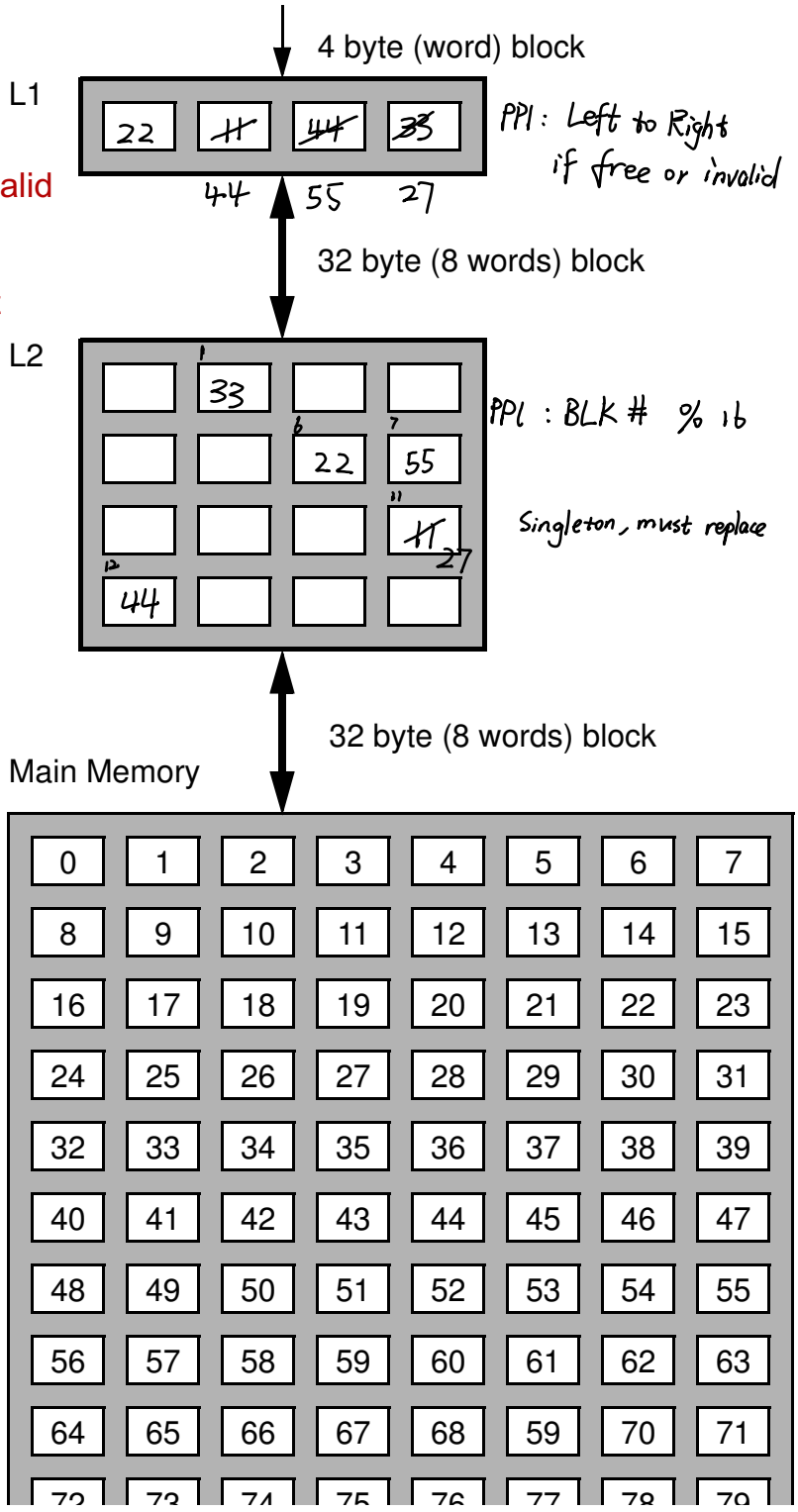
victim block

Block in cache that is chosen to be replaced.

working set

A group of blocks used by the code currently running.

~~*latency*~~ last pdf



Designing a Cache - Blocks

- *Cache memory is much, much smaller than the address space of the processor.*

  → Implications?

  Different blocks map to same location in cache.

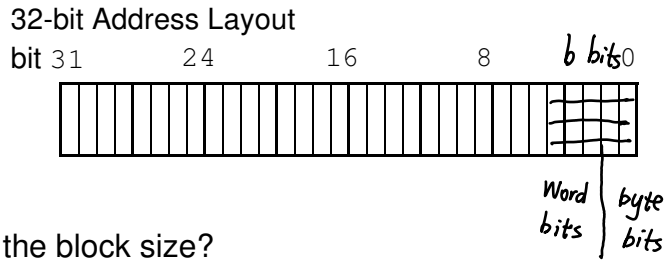
How many bytes of memory in an address space?

Let M be , The number of bytes in an address space, for IA32 4GB.
m bits: identify which bytes in the address space.
m = $\log_2 M$, if $M = 4GB$ then $m = 32$

- *Cache blocks must be big enough* to capture spacial locality, *but small enough to minimize* latency.

How big is a block?

Let B be the number of bytes per block.
IA32 32 bytes, IA64 64 bytes



→ How many bits of an address are needed for the block size?

b bits : identifies which byte in the block
b = $\log_2 B$, if $B = 32$ Then $b = 5$ bits

→ Which side of the address layout should be used to capture spatial locality in blocks?

Right side, least significant bits.

→ How do we find a particular word in a block?

Divide b bits
- word bits : identify which word block
- byte within word bits : identify which byte in word

How many 32 byte blocks of memory in a 32-bit address space?

$2^{(m-b)} = 2^{32-5} = 2^{27}$ 128 M 32 Byte block in address space

How many blocks in a cache?

cache size	number of 32 byte blocks
32KB	1 k
256KB	8 k
8MB	256 k

➢ How many blocks results in an effective cache???

Designing a Cache - Sets & Tags

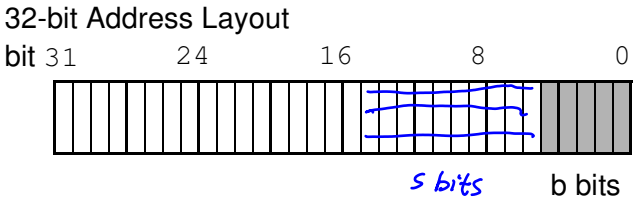
❋ *Using a random placement policy makes finding a block in the cache SLOW!*

→ Why? You must search the cache.
Better to limit where the block can go to a subset
of cache blocks (restricted placement policy)

How many sets in the cache?

Let S be the number of sets in the cache

s bits : identify which set in cache



$S \quad s = \log_2 S$

1024 10

8192 13

→ Which bits of an address should be used capture spatial locality for sets?

bits next to b bits ensure neighboring blocks
go to neighboring cache sets

→ How many blocks of memory map to each set for a 32-bit AS with 1024 sets? 8192?

$m - b - s$

1024 $32 - 5 - 10 = 17$ so $2^{17} = 128k$

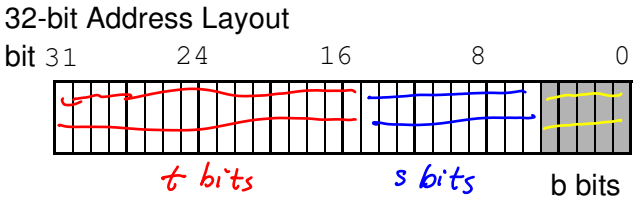
8192 $32 - 5 - 13 = 14$ $2^{14} = 16k$

Implications?

we need a way to determine if the block in cache set is the right one

3.) How do we identify which block is in a set?

use the remaining bits for a unique tag



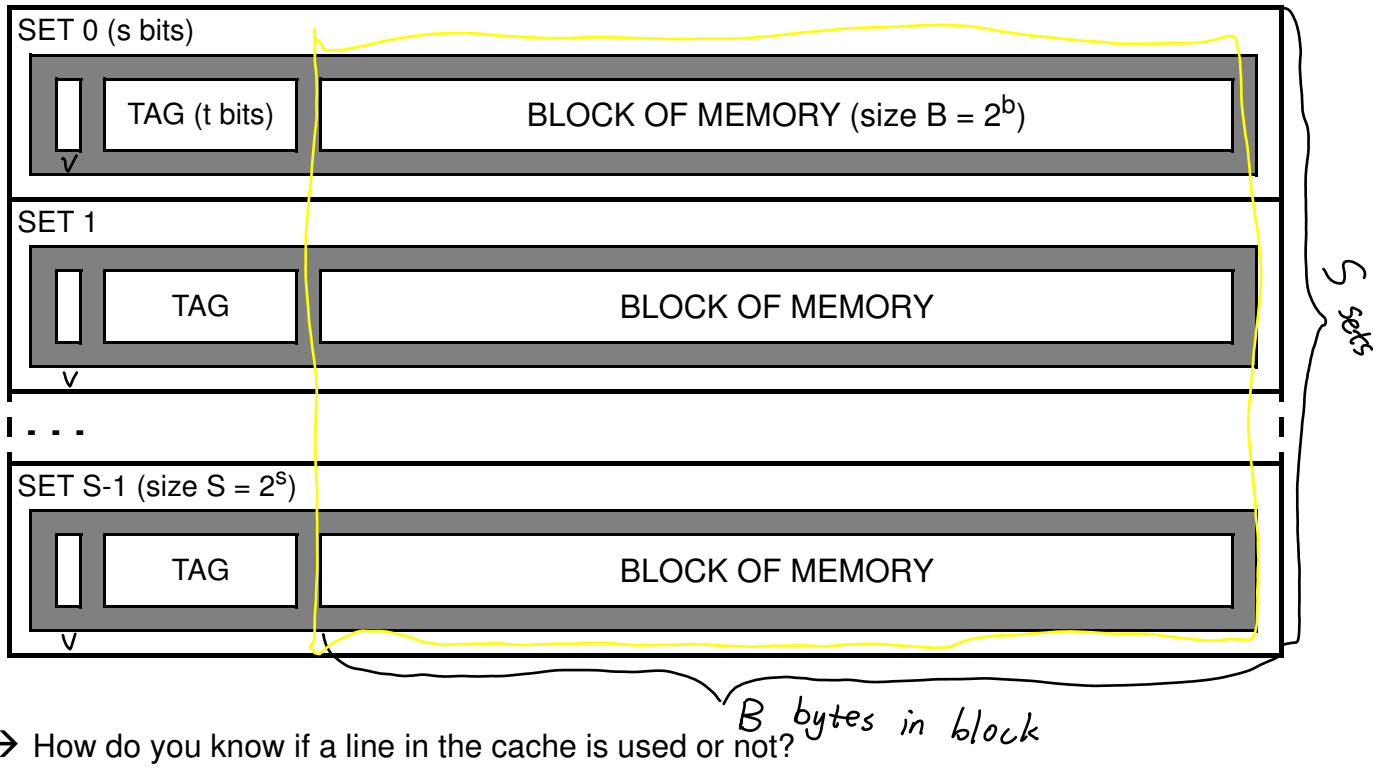
$t = m - b - s$

Designing a Cache - Lines

What? A *line* is

- one location in a cache that can store one memory block
- composed of storage for the memory block an info needed for the cache to function.

Basic Cache Diagram



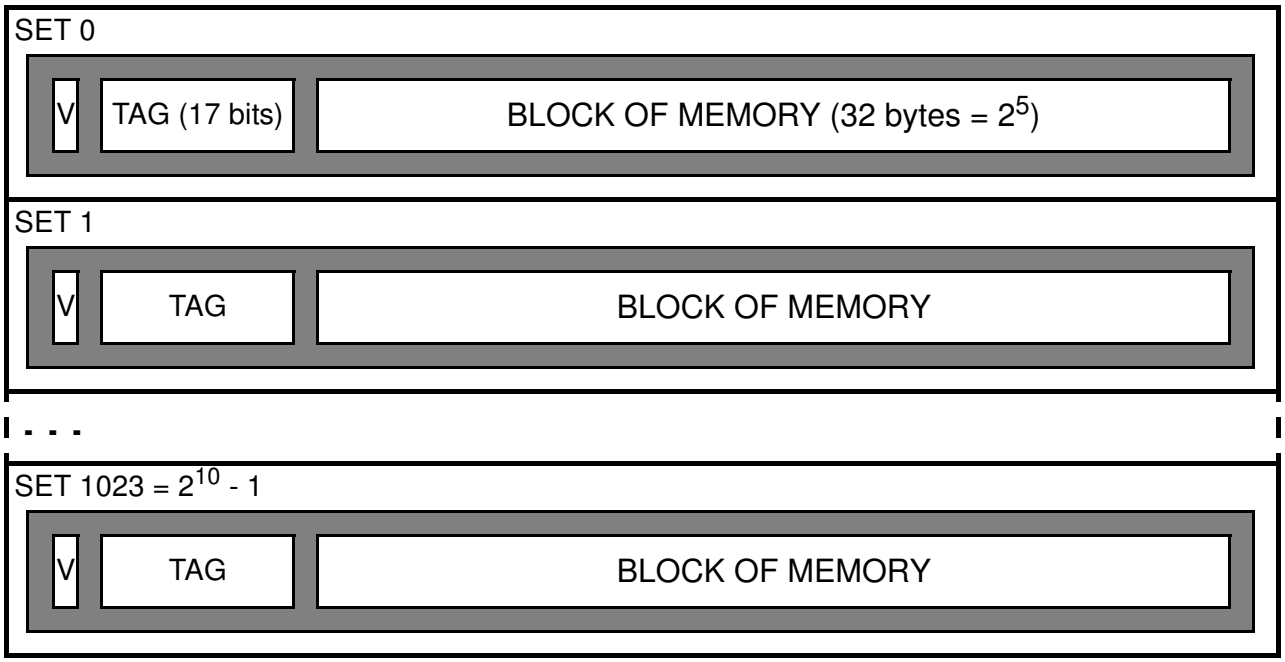
→ How do you know if a line in the cache is used or not?

use a valid bit
0 means not valid (free or invalid)
1 means valid (has usable info)

→ How big is a cache given S sets with blocks having B bytes?

Basic Cache Operation

Basic Cache Diagram



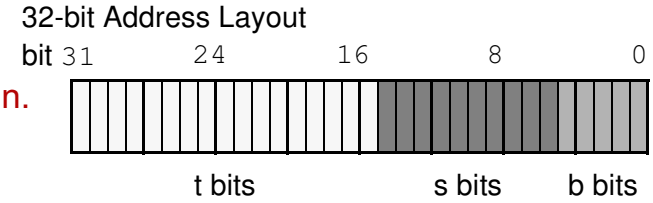
→ How big is this cache?

$SB = 1k \times 32 = 32k$

How does a cache process a request for a word at a particular address?

1. Set Selection

   extract s bits to identify which set to look in.



2. Line Matching

   extract t bits and compare with tag in the cache set.

   if no match or V bit is 0

   if match and V bit is 1