CS 354 - Machine Organization & Programming
Tuesday, October 22, 2019

Project p3 (6%): DUE at 10 pm on Monday, October 28th

Last Time

Memory Hierarchy
Locality
Bad Locality
Rethinking Addressing
Caching Basic Idea

Today

Caching Basic Idea (from last time)
Designing a Cache - Blocks
Designing a Cache - Sets and Tags
Basic Cache Lines
Basic Cache Operation
Basic Cache Practice
Direct Mapped Cache
Set Associative Cache

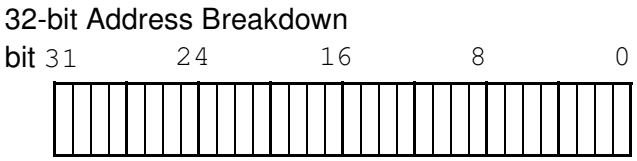
Next Time

Cache Performance and Coding Considerations
Read: B&O 6.4.5 - 6.4.7, 6.5 - 6.7

Designing a Cache - Blocks

❋ *The bits of an address*

How many bytes in an address space?



$M = 2^m$
$m = \log_2 M$

Let M be

Thus m is

How big is a block?

$B = 2^b$
$b = \log_2 B$

Let B be,

b bits:

- What is the problem with using the most significant bits (left side) for the b bits?

❋ *Cache blocks must be big enough*
but small enough

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

❋ *The remaining bits of an address*

Designing a Cache - Sets & Tags

❋ *A cache must be searched*

→ Problem?

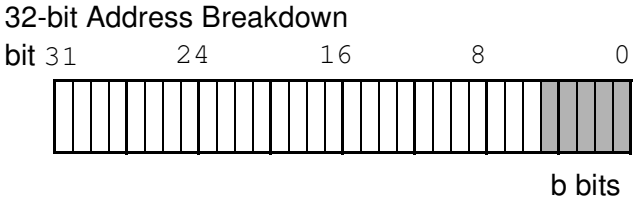
Improvement?

set:

How many sets in the cache?

$S = 2^s$
$s = \log_2 S$

Let S be



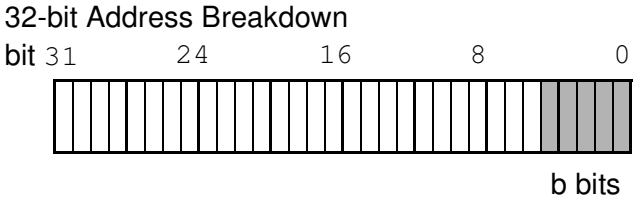
s bits:

➢ What is the problem with using the most significant bits (left side) for the s bits?

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

❋ *Different blocks of memory that map to same set*

Since different blocks map to the same set how do we identify which block is in a set?



t bits:

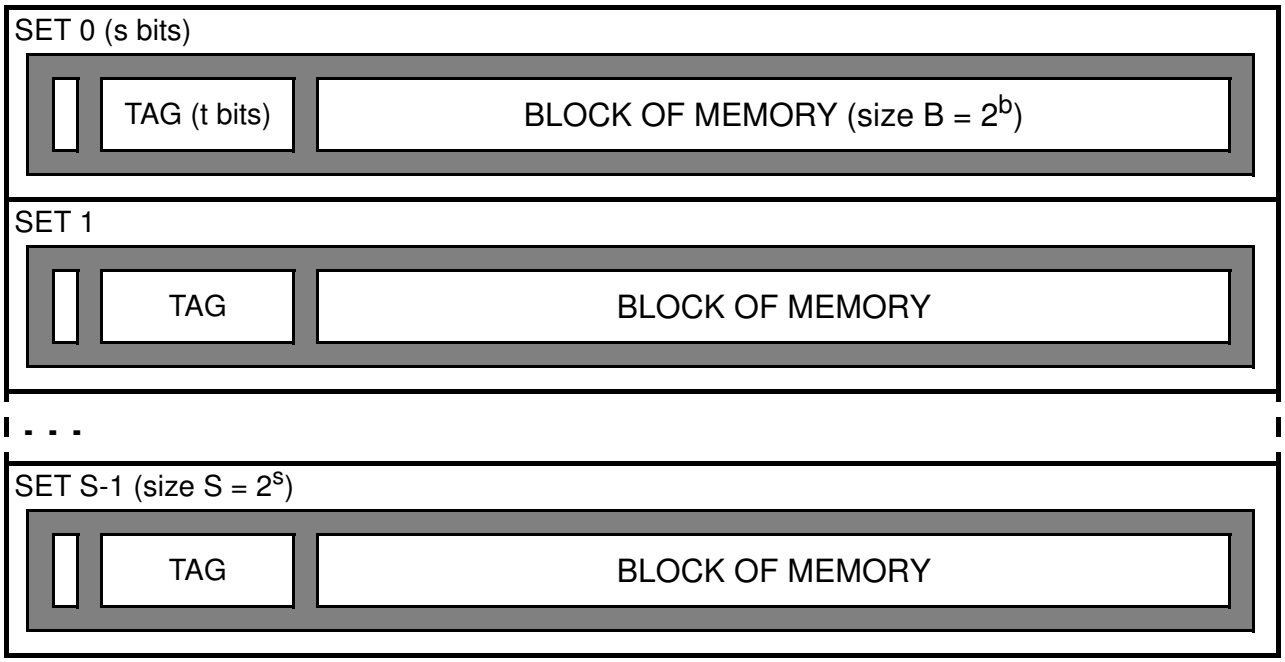
Basic Cache Lines

What? A *line* is

-
-

❋ *In our basic cache each set*

Basic Cache Diagram

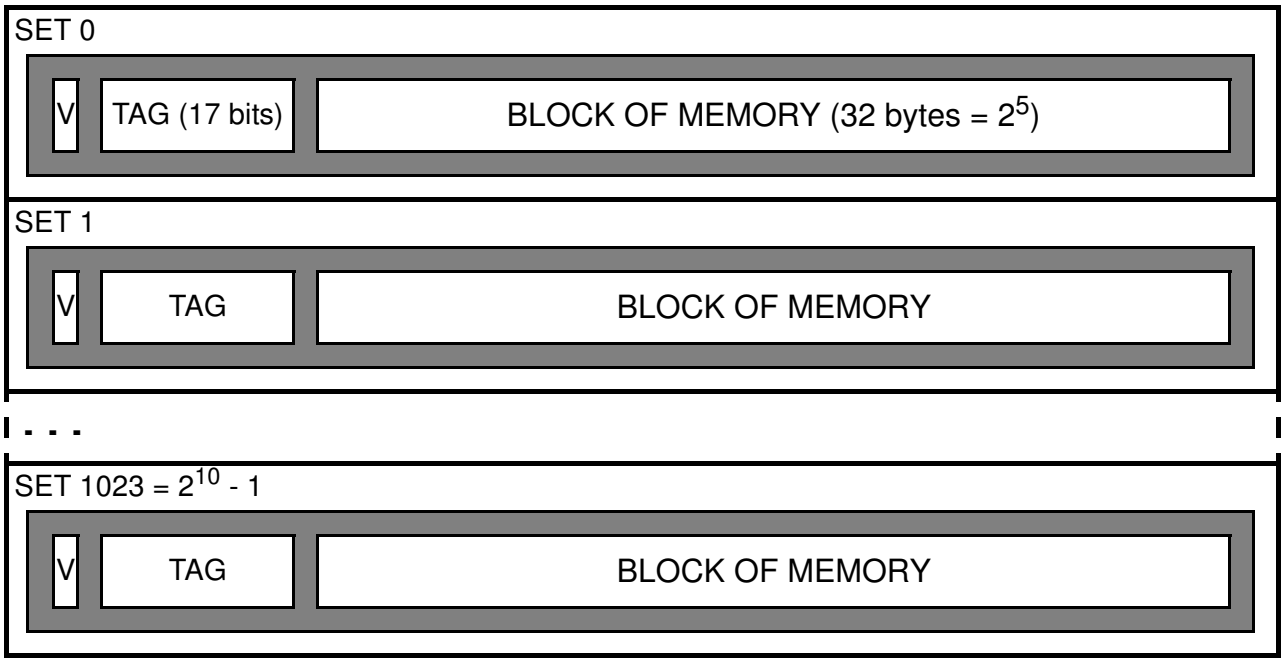


→ How do you know if a line in the cache is used or not?

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

Basic Cache Operation

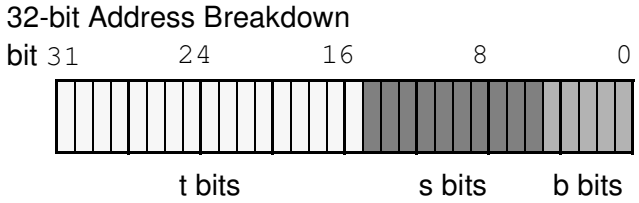
Basic Cache Diagram



→ How big is this basic cache?

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

1. Set Selection



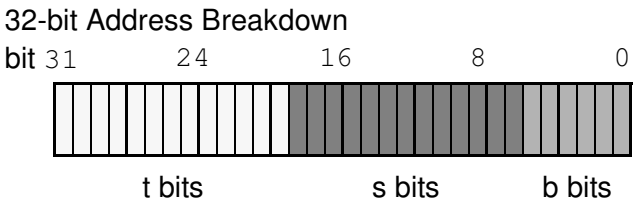
2. Line Matching

if no match or valid bit is 0

if match and valid bit is 1

Basic Cache Practice

You are given the following 32-bit address breakdown used by a cache:



→ How big are the blocks?

→ How many sets?

→ How big is this basic cache?

Assume the cache design above is given the following specific address: `0x07515E2B`

→ Which set should be checked given the address above?

→ Which word in the block does the L1 cache access for the address?

➢ Which byte in the word does the address specify?

→ If a set had the following V bit and tag, does the address produce a hit or miss?

	V	tag
1.)	1	0x0750
2.)	0	0x0750
3.)	1	0x00EA
4.)	0	0x00EA

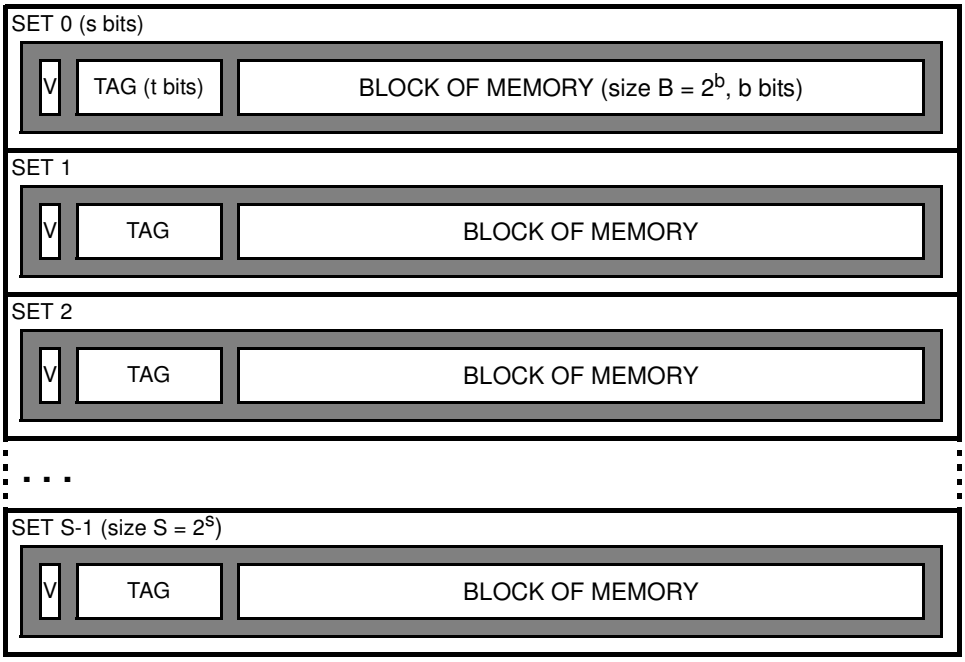
Direct Mapped Cache

Direct Mapped Cache

is a cache

+

+



→ What happens when two different memory blocks map to the same set?

→ Improvements?

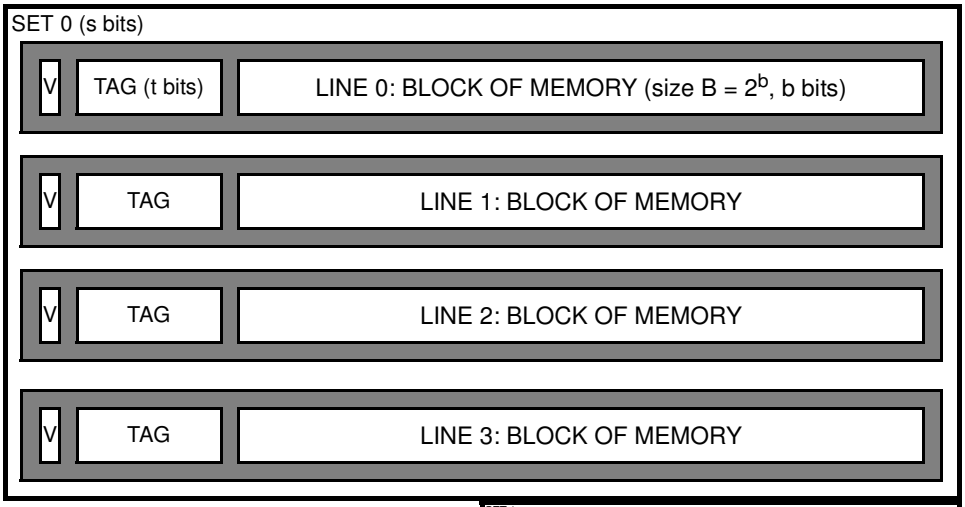
Set Associative Cache

Set Associative Cache

is a cache

+

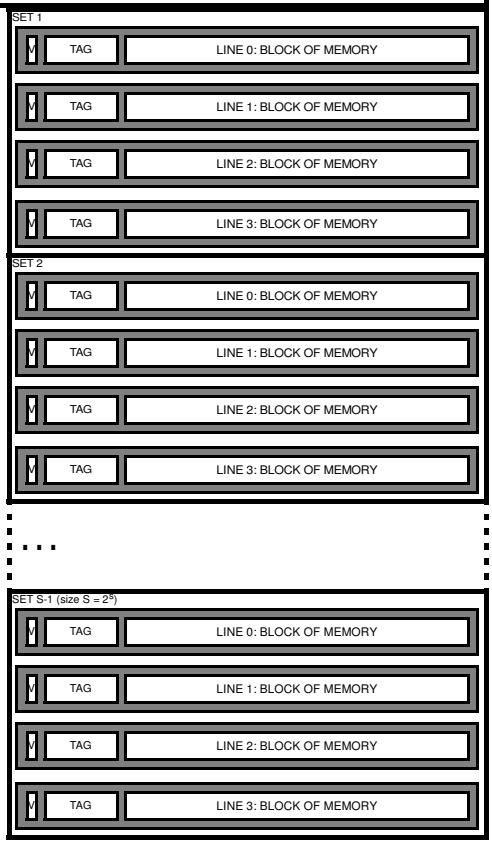
-



Let E be

E = 4 is

E = 1 is



Operation

1. Set Selection
2. Line Matching

❋ ***C = (S, E, B, m)***

Let C be

→ How big is a cache given (1024, 4, 32, 32)?