Memory Caches

CS520
Dept. of Computer Science
Univ. of New Hampshire
Big Picture

CPU

Cache

Memory

Locality

Temporal

Spatial
cache is a sequence of sets

each set contains lines

each line is a block plus control information

each block contains contiguous bytes/words from memory spatial locality
address

| tag | set index | offset |

used to check if block containing address is in line

select set

into block
Control information in line

Valid flag - does cache line contain a block?

Tag - upper bits for addresses of words/bytes in the block in the line

dirty flag - has block been modified since it was loaded into cache?

(used for "write-back" caches)

LRU stamp - used to implement/approximate Least Recently Used replacement policy

temporal
16-bit address

Address:

```
   | 1111 0000 10  | 1100  | 01 |
   |               |       |    |
   | 12 10         |       | 10 |
```

Block size = 4 bytes
16 sets
1 line per set

Diagram:

```
  Valid | tag  | block
  0     |      |      
  1     |      |      
  12    |      |      
```

Cache hit
memory is much larger than cache so multiple blocks will map to the same set use tags to distinguish

Caches with one line per set are known as direct-mapped
set-associative caches

multiple lines per set
must search lines in set to find matching tag
provides more flexibility when placing blocks into cache
Fully-associative caches

only 1 set

so blocks can be placed anywhere
in cache

most flexible

search for tag can now be expensive
replacement policy

when bringing new block into cache, will need to choose block to be replaced if set is full

usually LRU or approximation of LRU

or flag that is set when block is accessed and periodically cleared in all blocks in set
writes

if not in cache already, do memory read first
then modify block in cache

write-through:
modify cache and write change to memory (need a buffer for efficiency)

write-back:
accumulate changes in block in cache
utilize "dirty" flags
at replacement time write block to memory if "dirty" flag set
what if multiple CPUs sharing memory? 
with each CPU having its own cache 
need a method to ensure consistency across caches 
e.g. write to a block in one cache may cause invalidation of a copy of the block in another cache