Must be page 2

16 block cache
	Same cache location twice?
	Smallest direct mapped cache that will not use the
For the following access pattern, what is the

Review Problem 39

40011 2116 I 524
How many total bits are required for a direct-mapped cache with 64 KB of data and 16-bit block addresses? Assuming a 32-bit address, how many total bits are required?
<table>
<thead>
<tr>
<th></th>
<th>Data</th>
<th>Move Ales &amp; Chunk</th>
<th>Valid Bit</th>
</tr>
</thead>
</table>

Take advantage of spatial locality & amortize tag memory via larger block size.

Significant area overhead for tag memory.

Does not take advantage of spatial locality.

Uses temporal locality by holding on to previously used values.

Previous discussion assumed direct-mapped cache 1 byte blocks.

**Cache Block Overhead**
Assume 2^10 line direct mapped cache with 2^9 byte blocks.
$1200 / 16 = 75.0$

Block address = 3,$row 3, col 5 = C53 = Address of byte 5 of block 4.

$1200 \% 16 = 11.0$

$1200 \% 16 = 1200 \% 16 = 0$

$BS = ABCD 3:07 = 1200.0$ 16 = 0

$CI = ABCD 7:47 = 1200/16(16 \times 64) = 1.0$

$TAG = ABCD 31:10 7 = (1200/16 \times 64) = 1.0$

Given a cache with 64 blocks and a block size of 16 bytes, what block number does byte address 1200\textsuperscript{10} map to?

Cache Block Example $2^2$
If block size is too big relative to cache size, miss rate will go up:

- Takes longer time to fill up the block
- Larger block size means larger miss penalty

In general, larger block size takes advantage of spatial locality BUT:

Too few cache blocks
Threshing - Continually loading into cache but evicting it before reuse

What if regularly used items happen to map to the same cache line?

Direct Mapped Cache Problems