CS61C Fall 2012 – 5 – Caches

<u>Caches</u>

Conceptual Questions: Why do we cache? What is the end result of our caching, in terms of capability?

To make memory seem faster.

What are temporal and spatial locality? Give high level examples in software of when these occur.

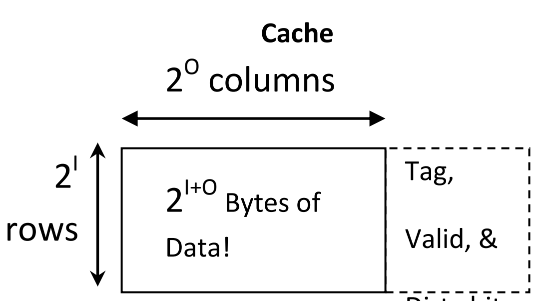
Temporal locality — if a value is accessed; it is likely to be accessed again soon
Examples: loop indices, accumulators, local variables in functions
Spatial locality — if a value is accessed; values near to it are likely to be accessed again soon
Examples: iterating through an array

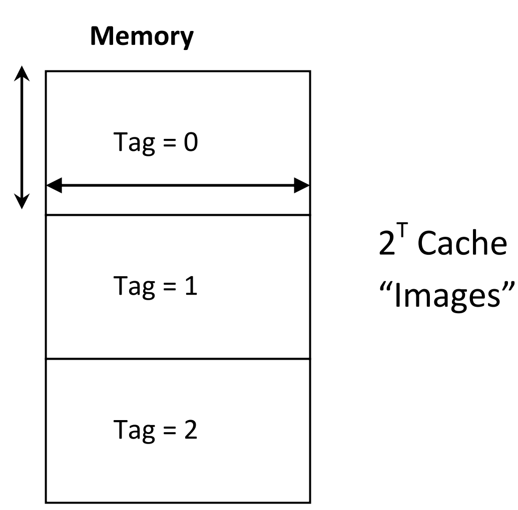
Break up an address:

Tag	**I**ndex	**O**ffset

Offset: “column index”, Indexes into a block. (O bits)
Index: “row index,” Indexes blocks in the cache. (I bits)
Tag: Where from memory did the block come from? (T bits)

Segmenting the address into TIO implies a geometrical structure (and size) on our cache. Draw memory with that same geometry!





Cache Vocab:
- **Cache hit** – Correct item is found and we write to the cache directly.
- **Cache miss** – Nothing in checked cache block, so read from memory and write to cache.
- **Cache miss, block replacement** – The right block was found, but it had the wrong tag. Do above.

Assume a write-through policy, fill out the table:

Address Bits	Cache Size	Block Size	Tag Bits	Index Bits	Offset Bits	Bits per Row
16	16KB	1B	**2**	**14**	**0**	**11**
16	16KB	16KB	**2**	**0**	**14**	**2^17 + 3**
16	16KB	**8B**	**2**	**11**	3	**67**
32	32KB	**8B**	**17**	12	**3**	**82**
32	64KB	**16B**	16	**12**	**4**	145
32	**512KB**	32B	**13**	**14**	**5**	270
64	**4MB**	**256B**	42	14	8	**2091**

Assume 16 B of memory and an 8B direct-mapped cache with 2-byte blocks. Classify each of the following byte-addr. memory accesses as hit (H), miss (M), or miss with replacement (R).

a. 0 **M**
b. 4 **M**
c. 1 **H**
d. 1 **H**
e. 10 **M**
f. 12 **R**
g. 0 **H**
h. 4 **R**

You want your AMAT to be <= 2 cycles. You have two levels of cache.

L1 hit time is 1 cycle. L1 miss rate is 20%

L2 hit time is 4 cycles L2 miss penalty is 150 cycles

What does your L2 miss rate need to be?

AMAT = Hit time + L1 Miss rate*(L2 Hit time + L2 Miss rate*L2 Miss penalty)

$2 >= 1 + .2(4 + 150x)$; $x <= .0066 = 0.66\%$

You know you have 1 MiB of memory (maxed out for processor address size) and a 16 KiB cache (data size only, not counting extra bits) with 1 KiB blocks.

```
#define NUM_INTS 8192
int A[NUM_INTS]; // lives at 0x100000
int i, total = 0;
for (i = 0; i < NUM_INTS; i += 128) A[i] = i; // Line 1
for (i = 0; i < NUM_INTS; i += 128) total += A[i]; // Line 2
```

a) What is the T:I:O breakup for the cache (assuming byte addressing)? **6:4:10**

b) Calculate the hit percentage for the cache for the line marked "Line 1". **50%**

c) Calculate the hit percentage for the cache for the line marked "Line 2". **50%**

How could you optimize the computation? **You could do the second loop in the opposite direction, or you could collapse the two loops into one.**