Caches

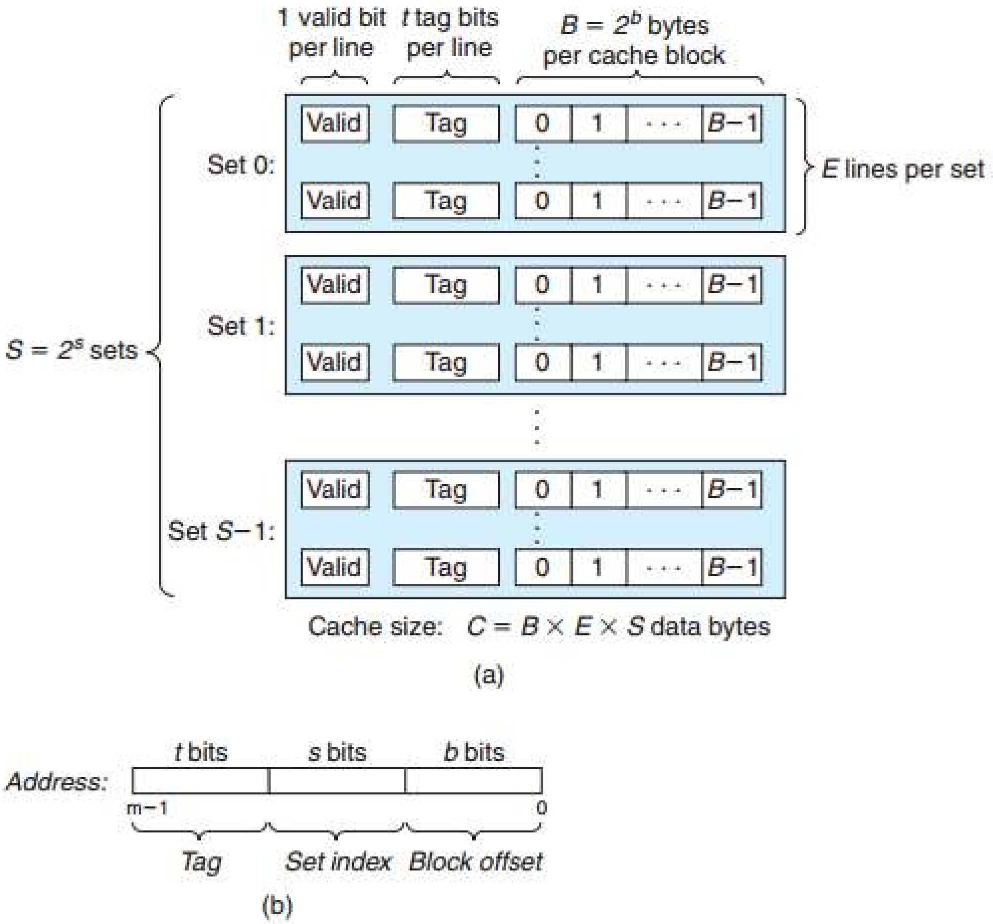
CSE 471 Spring 2015
Mark Wyse
April 16, 2015

Cache Basics

- Why do caches work?
  - Locality
- (S, E, B, m)
  - S = Sets
  - E = Associativity (Ways)
  - B = Block Size (Line size)
  - m address bits
- Cache Equation
  - C = S x E x B
- See CS:APP §6.4 (CSE 351) for a full refresher

Figure 6.27

General organization of cache (S, E, B, m). (a) A cache is an array of sets. Each set contains one or more lines. Each line contains a valid bit, some tag bits, and a block of data. (b) The cache organization induces a partition of the m address bits into t tag bits, s set index bits, and b block offset bits.



Victim Cache/Buffer

- What happens to evicted cache lines?
  - If dirty, write back to lower level cache
  - Overwrite
- But, what if we want to access a just evicted line?
  - Fetch from lower level cache….
  - Or, just cache the cache!
- Victim Cache: small fully associative cache, between L1 and L2 (lower level memory)
  - On L1 miss, check Victim Cache
  - If line found, swap L1 line and Victim line

Homework #2 – Mystery Caches

- Goal: discover cache parameters
  - MAX_* variables: hard maximums on different cache parameters
- Python
  - Basically pseudo-code that executes
  - Google / Stack Overflow are your friends
- Files
  - caches.py: defines a cache using LRU – don't modify
  - discover_cache_params.py: defines discovery routines – modify!
    - Should be self-explanatory, fill-in the # todo's

Partners

- You must work in pairs for this assignment (one solo)
- Everyone must fill out catalyst survey entering your partner’s name! (solo, enter your own name)