Zachary Seymour
CS 520
Assignment 3
November 22, 2013

1 Consider the following stream of memory word addresses directed at the cache. Assume that the memory is word addressable, and the total cache capacity is 64 words. Further assume that the cache is 2-way set-associative, the cache line size is 8 words and the LRU replacement policy is used.

$$5, 7, 1, 12, 34, 39, 65, 3, 107$$

For each of these accesses, indicate whether it is a cache hit or a cache miss, show how you computed the cache index, and depict the contents of the cache after every access. You do not have to show the tag bits, only show the addresses of the cached data.

Since our total size is 64 words and each cache line is 8 words, we have 8 cache entries and thus four sets in a 2-way set-associative cache. Thus, we proceed with the stream of addresses

- 5
  - Cache Index: $5 \mod 4 = 1$
  - Cache Miss
  - Cache contents

Set	Data	Data
0		
1	5	
2		
3		

- 7
  - Cache Index: $7 \mod 4 = 3$
  - Cache Miss
  - Cache contents

Set	Data	Data
0		
1	5	
2		
3	7	

- 1
  - Cache Index: $1 \mod 4 = 1$

- Cache Miss
- Cache contents

Set	Data	Data
0		
1	5	1
2		
3	7	

- 12
  - Cache Index: $12 \mod 4 = 0$
  - Cache Miss
  - Cache contents

Set	Data	Data
0	12	
1	5	1
2		
3	7	

- 34
  - Cache Index: $34 \mod 4 = 2$
  - Cache Miss
  - Cache contents

Set	Data	Data
0	12	
1	5	1
2	34	
3	7	

- 39
  - Cache Index: $39 \mod 4 = 3$
  - Cache Miss
  - Cache contents

Set	Data	Data
0	12	
1	5	1
2	34	
3	7	39

- 65
  - Cache Index: $65 \mod 4 = 1$
  - Cache Miss

- Cache contents

Set	Data	Data
0	12	
1	65	1
2	34	
3	7	39

- 3
  - Cache Index: $3 \mod 4 = 3$
  - Cache Miss
  - Cache contents

Set	Data	Data
0	12	
1	65	1
2	34	
3	3	39

- 107
  - Cache Index: $107 \mod 4 = 3$
  - Cache Miss
  - Cache contents

Set	Data	Data
0	12	
1	65	1
2	34	
3	3	107

2 Consider a system with physically-addressed caches, and assume that 40-bit virtual addresses and 32-bit physical addresses are used, and the memory is byte-addressable. Further assume that the cache is 4-way set-associative, the cache line size is 64 Bytes and the total size of the cache is 64 KBytes. Answer the following questions, providing adequate explanations in all cases:

2a What should be the minimum page size in this system to allow for the overlap of the TLB access and the cache access?

In this scenario, we have 1024 cache entries and thus 256 cache sets, which will require 7 bits for indexing. Further, our virtual address space is $2^{40} = 128$ GB We also know that we have $40 - (7 + 2) = 31$ bits remaining for our page number, so we have 2^{31} pages with 2^9 bits or 64 bytes as our minimum page size.

2b Repeat part (a) assuming that the cache associativity is increased to 8. Assume that the total cache size and the cache line size remain the same.

Here now, we still have 1024 cache entries, but only 128 sets, so we need 6 bits to index the cache. So, our page number increases to 32 bits, and we then have 2^8 bits or 32 bytes for our page size.

2c Assuming that the memory page size in this system is as calculated in your answer to Part (b), compute the total size of the page table in bytes. Assume that a simple linear page table is used and only the page translation information is stored in each entry, with no additional bits.

If we allocate $6 + 2 = 8$ bits for the page offset, then we have remaining 32 bits for the virtual page number, so we have 2^{32} pages at 32. We need $32 + 24 = 56$ bits per page, so the total size of the page table is 28 GB.

2d Repeat part (b) assuming that the OS can guarantee that the two least significant bits of the page number would not change during the address translation.

In this case, we use 29 bits for our page number, so we have 2^{11} bits or 256 bytes as our page size.

2e Assume that the OS needs to establish a mapping for the virtual page number 52356 (expressed in decimal). List some possible frame numbers in the physical memory where this page can be mapped to support the OS guarantee described in part (d).

$52356 = 0b1100110010000100$, so 52364, 52236, or 52748, for example, as long as the last two bits are still 00.

3 Consider the use of multi-level hot-cold bits in implementing a replacement algorithm for a set-associative cache with 2^p ways and S sets. This is an approximation of the LRU policy that marks the most recently assessed way as hot, and the other one as cold (for a 2-way set-associative cache). This mechanism can be extended to more than two ways by providing multiple levels of hot-cold bits. For example, for a 4-way cache the victim is determined as the cold way of the cold pair.

3a What is the total number of hot-cold bits for each set in this cache?

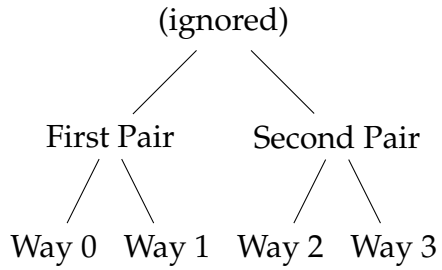
We can model the multi-level hot-cold bits as a binary tree, where the height corresponds to the log of the associativity, minus the root node, because a cache of associativity one will not need a hot-cold bit. So, the number of bits per set is $2^{p+1} - 2$.

3b What is the total number of hot-cold bits in the cache?

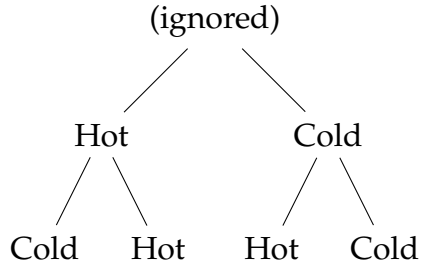
The total number of hot-cold bits in the whole cache is equal to the number per set times the number of sets, so we have $S\left(2^{p+1}-2\right)$.

3c Assume that $p = 2$ and accesses to a specific set result to the following ways in sequence and no misses occur:
Way 0, Way 3, Way 2, Way 0, Way 1, Way 1, Way 2, Way 3, Way 2, Way 0
At the end of the fifth access in this sequence (to Way 1), what are the settings of the hot-cold bit? Do they point to the same victim that would have been selected by a true-LRU algorithm?

Say we number our bits as such:



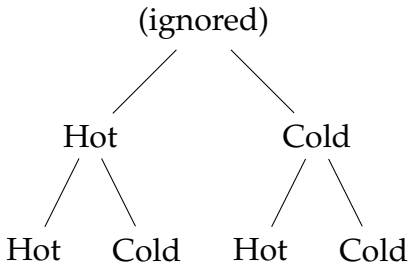
Bit	Hot/Cold
First Pair	~~1~~ ~~0~~ 1
Second Pair	~~0~~ ~~1~~ 0
Way 0	~~1~~ 0
Way 1	~~0~~ 1
Way 2	~~0~~ 1
Way 3	~~0~~ ~~1~~ 0



The victim pointed to then is Way 3, which is indeed the least recently used way as well.

3d Repeat Part (c), assuming that the all accesses shown in the sequence given in Part (c) have been completed with the last access being made to Way 0.

Bit	Hot/Cold
First Pair	~~1~~ ~~0~~ ~~1~~ ~~0~~ 1
Second Pair	~~0~~ ~~1~~ ~~0~~ ~~1~~ 0
Way 0	~~1~~ ~~0~~ 1
Way 1	~~0~~ ~~1~~ 0
Way 2	~~0~~ ~~1~~ ~~0~~ 1
Way 3	~~0~~ ~~1~~ ~~0~~ ~~1~~ 0



The victim pointed to then is again Way 3; however, now the least recently used way is Way 1.