Heather Warman and David Merrick
May 13, 2014
CS 472 HW 5

(9.2) Why do computers use cache memory?

Computers use cache memory to increase the performance of the CPU. To read the contents of the cache, the processor doesnt need to access the slower memory to read frequently used instructions or data.

(9.3) What is the meaning of the following terms?

a. Temporal locality

The memory address is accessed repeatedly within a short time span. One example of this is in a loop.

b. Spatial locality

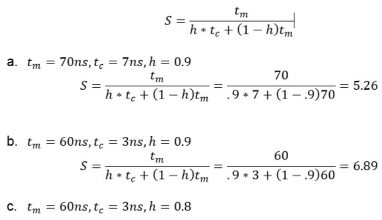
The memory address is clustered within the same region of memory. One example of this is a data structure.

(9.4) From first principles, derive an expression for the speedup ratio of a memory system with cache (assume the hit ratio is h and the ratio of the main storage access time to cache access time is k, where k ¡ 1). Assume that the system is an ideal system and that you dont have to worry about the effect of clock cycle times.

1/(h*k + (1-h)) = 1/(1-h(1-k))

(9.5) For the following ideal systems, calculate the speedup ratio S. In each case, t_c is the access time of the cache memory, t_m is the access time of the main store, and h is the hit ratio.

The speedup ratio is the ratio of the memory systems access time without cache to its access time with cache.



$$S = \frac{t_m}{h * t_c + (1-h)t_m}$$

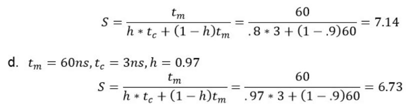
a. $t_m = 70ns, t_c = 7ns, h = 0.9$

$$S = \frac{t_m}{h * t_c + (1-h)t_m} = \frac{70}{.9 * 7 + (1-.9)70} = 5.26$$

b. $t_m = 60ns, t_c = 3ns, h = 0.9$

$$S = \frac{t_m}{h * t_c + (1-h)t_m} = \frac{60}{.9 * 3 + (1-.9)60} = 6.89$$

c. $t_m = 60ns, t_c = 3ns, h = 0.8$

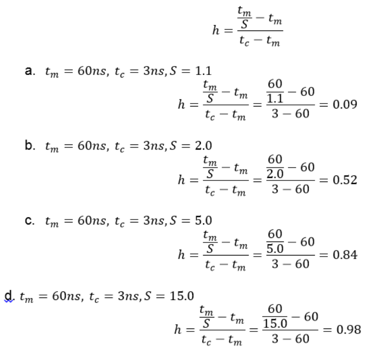


$$S = \frac{t_m}{h * t_c + (1-h)t_m} = \frac{60}{.8 * 3 + (1 - .9)60} = 7.14$$

d. $t_m = 60ns, t_c = 3ns, h = 0.97$

$$S = \frac{t_m}{h * t_c + (1-h)t_m} = \frac{60}{.97 * 3 + (1 - .9)60} = 6.73$$

(9.6) For the following ideal systems, calculate the hit ratio h required to achieve the stated speedup ratio S.



$$h = \frac{\frac{t_m}{S} - t_m}{t_c - t_m}$$

a. $t_m = 60ns,\ t_c = 3ns, S = 1.1$

$$h = \frac{\frac{t_m}{S} - t_m}{t_c - t_m} = \frac{\frac{60}{1.1} - 60}{3 - 60} = 0.09$$

b. $t_m = 60ns,\ t_c = 3ns, S = 2.0$

$$h = \frac{\frac{t_m}{S} - t_m}{t_c - t_m} = \frac{\frac{60}{2.0} - 60}{3 - 60} = 0.52$$

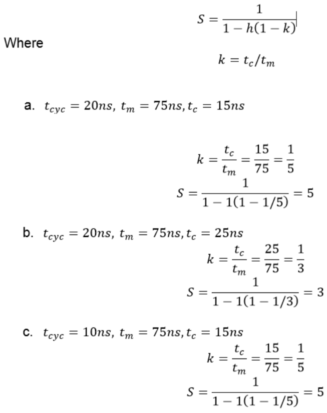
c. $t_m = 60ns,\ t_c = 3ns, S = 5.0$

$$h = \frac{\frac{t_m}{S} - t_m}{t_c - t_m} = \frac{\frac{60}{5.0} - 60}{3 - 60} = 0.84$$

d. $t_m = 60ns,\ t_c = 3ns, S = 15.0$

$$h = \frac{\frac{t_m}{S} - t_m}{t_c - t_m} = \frac{\frac{60}{15.0} - 60}{3 - 60} = 0.98$$

(9.8) For the following systems that use a clocked microprocessor, calculate the maximum speedup ratio you could expect to see as h approaches 100%.



$$S = \frac{1}{1 - h(1-k)}$$

Where

$$k = t_c/t_m$$

a. $t_{cyc} = 20ns,\ t_m = 75ns, t_c = 15ns$

$$k = \frac{t_c}{t_m} = \frac{15}{75} = \frac{1}{5}$$

$$S = \frac{1}{1 - 1(1 - 1/5)} = 5$$

b. $t_{cyc} = 20ns,\ t_m = 75ns, t_c = 25ns$

$$k = \frac{t_c}{t_m} = \frac{25}{75} = \frac{1}{3}$$

$$S = \frac{1}{1 - 1(1 - 1/3)} = 3$$

c. $t_{cyc} = 10ns,\ t_m = 75ns, t_c = 15ns$

$$k = \frac{t_c}{t_m} = \frac{15}{75} = \frac{1}{5}$$

$$S = \frac{1}{1 - 1(1 - 1/5)} = 5$$

(9.11) In a direct-mapped cache memory system, what is the meaning of the following terms?

a. Word

The word is the smallest unit of data in the cache. (There is a distinction to be made here between the smallest unit to be readable from the cache; if the processor only needed to read part of a word, it could read a word and ignore the unnecessary bits). A word in a direct-mapped cache is accessed by its set address and then its line address.

b. Line

A cache line is a sequence of several consecutive words. The line address selects the same line in each of the sets.

c. Set In a direct-mapped cache system, the lines are arranged into units called sets. The size of a set is the same size as the cache.

(9.12) How is data in main store mapped on to each of the following?

a. A direct-mapped cache

There is a direct relationship between the line in the cache and the location of the corresponding line in memory. There is only one location for each line. Data is mapped as follows: Set, Line, Word.

b. A fully associative cache

There are no restrictions on the locations of data, but it is necessary to choose which line to remove once the cache is full. Data is mapped as follows: Key, Index.

c. A set-associative cache

This type of cache combines the features of both fully associative and direct-mapped caches. It utilizes several direct-mapped caches in parallel. In an n-way set-associative cache, there are n possible cache locations that any given line can be loaded into. Data is mapped as follows: Set, Line, Word.

(9.17) What is cache coherency?

If the data in the cache is modified but is not changed in main memory (or vice-versa), the unchanged data is stale. This means inconsistency in the data. Cache coherency means that the data in the cache is consistent with the corresponding data in main memory.

(9.22) Why is it harder to design a data cache than an instruction cache?

It is more difficult to design a data cache than an instruction cache for a couple of reasons. First, an entry in an instruction cache is never modified, except when the line is initially moved into the cache. Second, the program does not change during the course of its execution, so instructions are never swapped out of the cache. In short, since the contents of the instruction cache are never changed it is much easier to design an instruction cache than a data cache.

(9.23) When a CPU writes to the cache, both the item in the cache and the corresponding item in the memory must be updated. If data is not in the cache, it must be fetched from memory and loaded in the cache. If t$_1 isthetimetakentoreloadthecacheonamiss, showthattheeffectiveaverageaccesstimeofthememorysystemisgivenbyt\_$
$ht\_c + (1 - h)t\_m + (1 - h)t\_1$.

h is the hit ratio.

t_c is the access time of the cache memory.

t_1 is the time taken to reload the cache on a miss.

t_m is the access time of the main store.

(1 - h) is the miss ratio.

So it makes sense that the average access time would be the probability of hitting the cache memory, plus the probability of hitting the main store, plus the probability of reloading the cache on a miss.

(9.26) A system has a level 1 cache and a level 2 cache. The hit rate of the level 1 cache is 90%, and the hit rate of the level 2 cache is 80%. An access to level 1 cache requires one cycle, an access to level 2 cache requires four cycles, and an access to main memory requires 50 cycles. What is the average access time?

Miss rate of L1 cache = .1. Miss rate of L2 cache = .2.

access time_avg=1 cycle+.1(4 cycles+.2(50 cycles))= 2.4 cycles

(9.28) In the context of multilevel caches, what is the difference between a local miss rate and a global miss rate?

A multilevel cache is arranged so that the lowest level cache successively moves up the ladder following a cache miss (L1 -¿ L2 -¿ L3, et al. until missing data is located). A local miss rate refers to the number of misses at a given cache level divided by the total number of memory accesses at that cache level, while a global miss rate refers to the rate of misses at all cache levels divided by the total number of cache accesses.

(9.35) A 64-bit processor has an 8-MB, four-way set associative cache with 32-byte lines. How is the address arranged in terms of set, line, and offset bits?

The line is 16 bits, the offset is 5 bits (because each line is $2^5 = 32bytes), andthesetis43bits$.

(9.41) What are the fundamental differences between cache memory (as found in a CPU) and cache memory found in a hard disk drive?

The main difference between cache memory in a CPU and cache memory in a hard disk drive is the type of memory used. The memory in a CPU needs to perform quickly, so it is much faster than the memory found in a hard disk drive, where performance is not as critical.

(9.42) What are the differences between write-back and write-through caches, and what are the implications for system performance?

In a cache system with a write-back policy, a write operation to the main memory takes place only when a line in the cache is to be ejected. So main memory is not updated on each write to the cache. The line is written back to memory only when its flushed out of the cache by a read miss. In a cache system with a write-through policy, the main memory is updated at the same time as the cache is loaded. A write-back policy results in higher speed, but lower cache coherency. A write through policy results in lower speed, but greater cache coherency.

(9.43) A computer with a 32-bit address architecture has a memory management system with single-level 4 KB page tables. How much memory space must be devoted to the page tables?

Page offset: Since there are 4KB in a page, offset field must contain 12 bits ($2^12 = 4096$).

Physical page number size = total bits allocated for physical address - page offset = 32 - 12 = 20.

Number of page entries = bits for virtual page number - page offset = 32 - 12 = 20. So we need $2^20 * 20bitsforthepagetablesize.Whichisabout2.5MB.$

(9.45) Average number of cycles per instruction = .7 * 1 cycle + .15 * 2 cycles + .1 * 2 cycles +.05 * (2 cycles + 5 cycles for write-through) + (1 - .95) * 10 for cache miss = 2.05 cycles.

(9.46) Each array access: 2 cycles + 6 cycles + 50 cycles = 58 cycles.

Setting the variable x and reading y: 2 cycles + 2 cycles = 4 cycles.

Setting s and reading x and s: 2 cycles + 2 cycles + 2 cycles = 6 cycles.

So each time around the loop is 58 cycles + 4 cycles + 6 cycles = 68 cycles.

(9.57) A computer with a 24-bit address bus has a main memory of size 16 MB and a cache size of 64 KB. The word length is two bytes.

a. What is the address format for a direct-mapped cache with a line size of 32 words?

Sets: 16MB/64KB = 256 sets.

Set: x bits. Line: x bits. Word: 16 bits.

b. What is the address format for a fully associative cache with a line size of 32 words?

c. What is the address format for a four-way set-associative cache with a line size of 16 words?