Jacob Branaugh, Brenn Kucey
November 11, 2013
CS 472 HW 4

(9.2) Why do computers use cache memory?

- Computers use cache memory because it transfers data quickly, and has a much larger capacity than the processor's registers. Cache memory is slightly slower than registers, but is significantly faster than RAM. There are only a handful of registers available, whereas cache can range from kB to MB. RAM, however, is usually available in the order of GB.

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

- Temporal locality

- Temporal locality describes the number of times a memory address is accessed within some amount of time.

- Spatial locality

- Spatial locality is a term to describe the physical closeness of related memory addresses.

(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 don't have to worry about the effect of clock cycle times.

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

(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.

a) $t_m = 70ns, t_c = 7ns, h = .9$

$S = \frac{1}{1-.9(1-\frac{7}{70})} = \mathbf{5.263}$

b) $t_m = 60ns, t_c = 3ns, h = .9$

$S = \frac{1}{1-.9(1-\frac{3}{60})} = \mathbf{6.897}$

c) $t_m = 60ns, t_c = 3ns, h = .8$

$S = \frac{1}{1-.8(1-\frac{3}{60})} = \mathbf{4.167}$

d) $t_m = 60ns, t_c = 3ns, h = .97$

$S = \frac{1}{1-.97(1-\frac{3}{60})} = \mathbf{12.739}$

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

a) $t_m = 60ns, t_c = 3ns, S = 1.1$

$h = \frac{1-\frac{1}{S}}{1-\frac{t_c}{t_m}} = \frac{1-\frac{1}{1.1}}{1-\frac{3}{30}} = \mathbf{0.096}$

b) $t_m = 60ns, t_c = 3ns, S = 2.0$

$h = \frac{1-\frac{1}{2}}{1-\frac{3}{60}} = \mathbf{0.526}$

c) $t_m = 60ns, t_c = 3ns, S = 5.0$

$h = \frac{1-\frac{1}{5}}{1-\frac{3}{60}} = \mathbf{0.842}$

d) $t_m = 60ns, t_c = 3ns, S = 15.0$

$h = \frac{1-\frac{1}{15}}{1-\frac{3}{60}} = \mathbf{0.982}$

(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%.

$S = \frac{1}{1-1(1-\frac{t_c}{t_m})} = \frac{1}{\frac{t_c}{t_m}} = \frac{t_m}{t_c}$, where t_m *and* t_c must be integer multiples of the time taken for one clock cycle.

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

$t_m = 75ns \rightarrow t_m = 4\ cycles$
$t_c = 15ns \rightarrow t_c = 1\ cycle$
$S = \frac{4}{1} = \mathbf{4}$

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

$t_m = 75ns \rightarrow t_m = 4\ cycles$
$t_c = 25ns \rightarrow t_c = 2\ cycle$
$S = \frac{4}{2} = \mathbf{2}$

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

$t_m = 75ns \rightarrow t_m = 8\ cycles$
$t_c = 15ns \rightarrow t_c = 2\ cycle$
$S = \frac{8}{2} = \mathbf{4}$

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

- Word: the number of bytes a processor core can process during one clock cycle, usually no larger than 64 bits
- Line: the minimum amount of data transferable between cache and main memory at a time, equal to a small number of words
- Set: a collection of lines equal to $\frac{size\ of\ main\ memory}{size\ of\ cache}$

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

- Direct mapped cache: main memory is organized into sets, which are organized into lines (see 9.11)
- Fully associative cache: cache takes a line directly from main memory and each line is paired with a tag address location and a bit that stores the state of the data's validity. This is an expensive type of cache, as it requires physical space for both the data bits and the tag bits, resulting in either twice the cache size needed or half the cache size usable.
- Set associative cache: the cache contains multiple sets, where the address of any given line is the same across all sets. The sets are searched in parallel, and a match in any set causes the search to end.

(9.17) What is cache coherency?

- Cache coherency is the state of a piece of data in cache versus the state of that same piece of data in main memory. Since data living in the cache also exists in the main store, both copies must be updated when the processor modifies the data. If the data in one is updated but data in the other has not yet been updated, the old data may be grabbed and used further.

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

- Because an entry in the instruction cache is never modified outside of the initial swapping in. Additionally, the instructions that are overwritten in the I-cache don't matter anymore, as the program doesn't change during its execution.

(9.23) When a CPU writes to 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 is the time taken to reload the cache on a miss, show that the effective average access time of the memory system is given by

$t_{avg} = ht_c + (1-h)t_m + (1-h)t_1$

- The average access time assuming no cache reload is:
$t_{avg} = ht_c + (1-h)t_m$,
meaning the cache access time multiplied by the hit percentage plus the memory access time multiplied by the cache-miss percentage.

- Now, assuming the cache takes time to reload after miss, that time must simply be added to our existing access time, giving:
$t_{avg} = ht_c + (1-h)t_m + (1-h)t_1$,
where our access time is the cache-hit percentage multiplied by the time taken for cache hit plus cache-miss percentage multiplied by the time taken for memory access plus cache-miss percentage multiplied by the time taken to reload the cache on 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 store requires 50 cycles. What is the average access time?

- $t_{avg} = h_1 t_{c1} + (1-h_1)h_2 t_{c2} + (1-h_1)(1-h_2)t_m = .9 \times 1 + .1 \times .8 \times 4 + .1 \times .2 \times 50 =$ **2.22 cycles**

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

- A local miss rate is the miss rate of a given cache independent of the miss rates of the rest. The global miss rate is the miss rate of all present caches, or the complement of the total memory access rate.

(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?

- Two bits are for selecting the set, sixteen bits are for selecting the line in the set, and two bits are for selecting the offset in the line.

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

- CPU cache holds lines, hard disk cache holds pages.

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

- Write-through caching is when the main memory is updated at the same time as the cache is loaded. This is the lower performance method, as the write time to memory is much slower than write time to cache.

- Write-back caching is when main memory is updated only when the cache is going to eject a line, which will be written back to main memory. This is the higher performance method, as fewer slow memory operations will be taking place.

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

(9.45) A computer runs an instruction set with the characteristics in the following table:

Class	Instruction Frequency	Cycles
Arithmetic Operations	70%	1
Conditional Operations	15%	2
Load	10%	2
Store	5%	2
Hit Rate	95%	
Cost of Cache Miss (Read)		10
Write-Through Time (Writes to Memory Are Not Buffered)		5

What is the average number of cycles per instruction?

- $t_{avg} = f_{arith}t_{arith} + f_{cond}t_{cond} + f_{load}(h_{rate}t_{load} + (1 - h_{rate})t_{read}) + f_{store}t_{write} = .7 \times 1 + .15 \times 2 + .1(0.95 \times 2 + (1 - .95) \times 10) + .05 \times 5 =$**1.49**

(9.46) Consider the following code that accesses three values in memory scalar integers x and s, and an integer vector y[i]. What is the memory latency in clock cycles for a trip round the loop (after the first iteration)? Assume that the array is not cached and each new access to the array results in a miss. The system has both L1 and L2 caches. The access time of the L1 cache is two cycles, the access time of the L2 cache is 6 cycles, and main memory has an access time of 50 cycles. In this case all memory and cache memory access take place in parallel.

```
for (i = 0; i < 100; i++) {
        x = y[i];
        s = s + x;
}
```

- More math possibly?

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

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

- asdf

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

- asdf

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

- asdf