CSE320 Fall 2014

Homework#8
Quiz in Lecture Tues. Dec. 2, 2014
There are no make-up quizzes!

Problem 1: Consider a direct-mapped cache with 2^{16} words in main memory. The cache has 16 blocks of 8 words each. It is a **word-addressable computer**.

(a) How many blocks of main memory are there?

Since there are 2^{16} words of main memory and each cache block contains 8 words, there are $\frac{2^{16}}{8} = \frac{2^{16}}{2^3} = 2^{13}$ blocks of main memory.

(b) What is the format of a memory address as seen by the cache, that is, what are the sizes of the tag, cache block, and block offset fields (if they apply)?

Main memory only needs 16-bits to represent each address uniquely

16-bit word address: 9 bits tag (rest of the address), 4 bits cache block ID, 3 bits for block offset of each word in block

(c) To which cache block will the memory reference $DB63_{16}$ map?

1101101101100011_2, Tag = 110110110, Cache block = 1100, block offset = 011

Since the cache block is $1100_2 = 12_{10}$, the cache block is 12.

Problem 2: Consider a **byte-addressable** computer with 16MB of main memory, a cache capable of storing a total of 64KB of data and block size of 32 bytes.

(a) How many bits in the memory address?

16MB = $2^{4} * 2^{20} = 2^{24}$, Address is 24-bits.

(b) How many blocks are in the cache?

Cache is 64KB = $2^6 2^{10} = 2^{16}$; each block is 32bytes = 2^5; $2^{16}/2^5 = 2^{11}$ blocks in cache.

(c) Assume the cache is direct mapped. What is the format of the memory address, including names and sizes?

Tag 8-bits; Cache block 11-bits; Block Offset 5-bits

(d) Assume the cache is 4-way set associative. What is the format of the memory address, including names and sizes?

Each cache entry can hold 4 blocks, $2^{11}/2^2 = 2^9$

Tag 10-bits; Cache block 9-bits; Block Offset 5-bits

(e) Assume the cache is fully associative. What is the format of the memory address, including names and sizes?

Tag 19-bits; Block Offset 5-bits

Problem 3: Consider a **byte-addressable** computer, ie. You access things in bytes instead of words, with 16-bit addresses, a cache capable of storing a total of 2KB of data, and blocks of 8 bytes. What is the format of the memory address, including names and sizes, for (a) direct mapped, (b) fully associative,

(c) 4-way set associative cache? Where in the cache would the memory address BAD016 be mapped for each of the three mappings (answer in binary or decimal)?

Instructions are 16 bits long. Blocks are 8 bytes (2^3 bits needed to represent the items in each block). The cache is 2^{11} bytes.

(a) There are $2^{11}/2^3 = 2^8$ blocks in the cache. Direct mapped tag = 5bits, block = 8bits, word (offset) = 3bits

(b) fully associative tag = 13bits, word(offset)= 3bits

(c) Each "way" holds 4 blocks. Therefore, the cache can hodl $2^{11}/2^3 = 2^8$ blocks, and then each "way" holds 4 of these. Then there are $2^8/2^2 = 2^6$ cache sets (positions/rows).

4-way set associative tag= 7bits, index = 6bits, word(offset) = 3bits

$BAD0_{16}$ = 1011 1010 1101 0000 so we have:

(a) 10111 01011010 000 for direct mapped, which indicates block 90 of cache

(b) 1011101011010 000 for fully associative, which can be placed anywhere

(c) 1011101 011010 000 for 4-way set associative, which indicates set 26 of cache

Problem 4: Suppose we have a **byte addressable** computer that has a 32-byte cache with 8 bytes per block. The memory address is 8 bits long. The system accesses memory addresses (in hex) in this exact order: 6E, B9, 17, E0, 4E, 4F, 50, 91, A8, AB, AD, 93, and 94.

(a) Assuming the cache is direct mapped, what memory addresses will be in cache block 2 after the last address has been accessed?

90, 91, 92, 93, 94, 95, 96, 97

(b) Assuming the cache is direct mapped, what is the hit ratio for the entire memory reference sequence given, assuming the cache is initially empty?

Addr	Block Index	HIT/MISS
6E	1	MISS
B9	3	MISS
17	2	MISS
E0	0	MISS
4E	1	MISS
4F	1	HIT
50	2	MISS
91	2	MISS
A8	1	MISS
AB	1	HIT
AD	1	HIT
93	2	HIT
94	2	HIT

HIT RATIO = 5/13 = 38.4%

(c) Assuming the cache is 2-way set associative with a LRU replacement policy, what is the hit ratio?

Given the cache is 2-way set associative, there are only 2 cache blocks in the cache, with 2 entries (A and B) in each.

Addr	Block Index	HIT/MISS
6E	1A	MISS
B9	1B	MISS
17	0A	MISS
E0	0B	MISS
4E	1A	MISS
4F	1A	HIT
50	0A	MISS
91	0B	MISS
A8	1B	MISS
AB	1B	HIT
AD	1B	HIT
93	0B	HIT
94	0B	HIT

HIT RATIO = 5/13 = 38.4%

Problem 5: The computer spends 82% of the time computing and 18% waiting for the disk. The instruction mix and the average cycles per instruction (CPI) for each type is:

Type	Percentage	CPI
Integer	40%	1
Floating Point	30%	5
All Others	30%	2

(a) What is the speedup of the machine if the processor is replaced with a new one that reduces the total computation time by 35%.

The processor is replaced with one that reduces computation time by 35%. Speedup = 1 / ((1 - 0.82) + 0.82 * 0.65) = 1.40

(b) What is the speedup of the machine if the disk is replaced with a solid state device that reduces the disk waiting time by 85%.

The disk is replaced with one that reduces disk wait time by 85%. Speedup = 1 / ((1 - 0.18) + 0.18 * 0.15) = 1.18

(c) What is the speedup of the machine if the CPI of floating point operations was reduced to 1?

Average CPI (old) = 0.40 * 1 + 0.30 * 5 + 0.30 * 2 = 2.5
Average CPI (enhanced) = 0.40 * 1 + 0.30 * 1 + 0.30 * 2 = 1.3
Speedup (computation) = 2.5 / 1.3 = 1.9
Speedup = 1 / ((1 - 0.82) + 0.82 / 1.9) = 1.64

(d) Would an infinitely fast disk drive outperform the the speedup in parts (i-iii)?

If the disk was infinitely fast: Speedup = 1 / (1 - 0.18) = 1.22 $\leq$ still slower than part a

Problem 6: Calculate the performance of a processor taking into account stalls due to data cache and instruction cache misses.

- The data cache has a 92% hit rate with a 2 cycle hit latency and 124 cycle miss penalty (cache miss and memory latency together).
- The instruction cache has a 90% hit rate with a a 2 cycle hit latency and 50 cycle miss penalty.
- The base CPI of the processor is 1.0
- 30% of instructions are loads and stores.

Assume the load never stalls a dependent instruction and assume the processor must wait for stores to finish when they miss the cache. Finally, assume that instruction cache misses and data cache misses never occur at the same time. Show your work.

(a) Calculate the additional CPI due to the instruction cache stalls.

The additional CPI due to icache stalls = Hit Rate * Hit Latency + Miss Rate* Miss Penalty = 0.9*2 + 0.1*50 = 1.8 + 5 = 6.8

(b) Calculate the additional CPI due to the data cache stalls.

The additional CPI due to dcache stalls = 0.92*2 + 0.08*124 = 11.76

(c) Calculate the overall CPI for the machine.

The overall CPI = 0.3*11.76 + 0.7*1.0 + 1.0*6.8 = 11.03

Problem 7: Suppose a computer using set associative cache has 2^{21} words of main memory and a cache of 64 blocks, where each cache block contains 4 words. If the cache is 4-way set associative, what is the format of a memory address as seen by the cache?

The 64 blocks in the cache must be divided into sets with 4 blocks each, implying there are only 16 sets. The instruction length is 21 bits: 15 bits for the tag (remaining of 21), 4 bits for the index (16 sets), 2 bits for the word (since cache block is 4 words).

Problem 8: Consider two processors with different cache configurations:

- Cache 1: Direct Mapped with one-word blocks. Instruction miss rate is 4%, data miss rate 6%
- Cache 2: Two-way set associative with four-word blocks. Instruction miss rate is 2%, data miss rate 3%

For these processors, one-half of the instructions contain a data reference. Assume that the cache miss penalty is 6 + block size in words. The CPI for this workload is measured on the processor with Cache 1 and was 2.0.

(a) Determine which processor spends the most cycles on cache misses.

(b) If the cycle times are 420ps and 310ps for processor with Cache 1 and 2 respectively, determine which processor is the fastest.

(a) Memory-stall clock cycles = Instructions/ Program * Misses/Instruction * Miss penalty
Misses/Instruction = Instruction miss rate + (Data miss rate* Data references/Instruction)
Miss penalty = 6 + Block size in words
Data references/Instruction = 50%
Cache 1: Block size = 1 word
Miss penalty = 6+1 =7 cycles
Instruction miss rate = 4%
Data miss rate = 6%

CPI_{stall} = (Instruction miss rate + (Data miss rate* Data references/Instruction) * Miss penalty
= (4% + (6%*50%)) * 7 = 0.49

Cache 2:
Block size = 4 words
Miss penalty = 6+4 = 10 cycles
Instruction miss rate = 2%
Data miss rate = 3%

CPI_{stall} = (Instruction miss rate + (Data miss rate* Data references/Instruction) * Miss penalty
= (2% + (3%*50%)) * 10 = 0.35
So cache 1 spends the most cycles on cache misses.

(b) CPU time = IC * CPI * Clock cycle
Instruction count is same for all processors
CPI = $\text{CPI}_{ideal-cache}$ + Stall cycles per instruction
$\text{CPI}_{ideal-cache}$ is the same for all processors
For processor 1: CPU time = IC * ($\text{CPI}_{ideal-cache}$ + 0.49) * 420 ps
For processor 2: CPU time = IC * ($\text{CPI}_{ideal-cache}$ + 0.35) * 310 ps

If you assume IC = 100 and $\text{CPI}_{ideal-cache}$ = 2.0 in both cases.
For processor 1: CPU time = 100 * (2.49) * 420 ps = 104580ps
For processor 2: CPU time = 100 * (2.35) * 310 ps = 73780ps

Processor 2 is faster.