CS698Y-Modern Memory Systems **Name:** ____________

Tutorial 1

10th August, 2017

Time Limit: 60 Minutes **Roll No.:** ____________

1. You are trying to appreciate how important the principle of locality is in justifying the use of a cache memory, so you experiment with a computer having an L1 data cache and a main memory (you exclusively focus on data accesses). The latencies (in CPU cycles) of the different kinds of accesses are as follows: cache hit, 1 cycle; cache miss, 105 cycles; main memory access with cache disabled, 100 cycles.

   (a) When you run a program with an overall miss rate of 5%, what will the average memory access time (in CPU cycles) be?

   $0.95 * 1 + 0.05 * 105 = 6.2$ cycles

   (b) Next, you run a program specifically designed to produce completely random data addresses with no locality. Toward that end, you use an array of size 256 MB (all of it fits in the main memory). Accesses to random elements of this array are continuously made (using a uniform random number generator to generate the elements indices). If your data cache size is 64 KB, what will the average memory access time be?

   Hit rate $\approx \frac{64 \text{ KB}}{256 \text{ MB}} = 0.00025$

   A.M.A.T $= 0.00025 * 1 + (1 - 0.00025) * 105$

   $= 104.97$ cycles

(c) If you compare the result obtained in part (b) with the main memory access time when the cache is disabled, what can you conclude about the role of the principle of locality in justifying the use of cache memory?

Access time when cache disabled

= 100 cycles

which is less than (b)

So, cache will be of no use.

(d) You observed that a cache hit produces a gain of 99 cycles (1 cycle vs. 100), but it produces a loss of 5 cycles in the case of a miss (105 cycles vs. 100). In the general case, we can express these two quantities as G (gain) and L (loss). Using these two quantities (G and L), identify the highest miss rate after which the cache use would be disadvantageous.

Memory access time with no cache = T_{off}

with cache = T_{on}

miss rate = m

$$T_{on} = (1-m)(T_{off}-G) + m(T_{off}+L)$$

Cache becomes useless, when

$$T_{off} <= (1-m)(T_{off}-G) + m(T_{off}+L)$$

$$m \geq \left(\frac{G}{G+L}\right) \geq \left(\frac{99}{104}\right)$$

~ 0.95

2. Increasing a cache's associativity (with all other parameters kept constant), statistically reduces the miss rate. However, there can be pathological cases where increasing a cache's associativity would increase the miss rate for a particular workload. Consider the case of direct mapped compared to a two-way set associative cache of equal size.

Assume that the set associative cache uses the LRU replacement policy. To simplify, assume that the block size is one word. Now construct a trace of word accesses that would produce more misses in the two-way associative cache. (Hint: Focus on constructing a trace of accesses that are exclusively directed to a single set of the two-way set associative cache, such that the same trace would exclusively access two blocks in the direct-mapped cache.)

Trace of $(T_1, T_2, T_3)^{\infty}$

3. Memory Hierarchy.

(a) Assume that we have a 32-bit processor (with 32-bit words) and that this processor is byte-addressed (i.e. addresses specify bytes). Suppose that it has a 512-byte cache that is two-way set-associative, has 4-word cache lines, and uses LRU replacement. Split the 32-bit address into "tag", "index", and "cache-line offset" pieces. Which address bits comprise each piece?

Cache-line-offset = 4 bits (0 to 3)
(4 words and each word is of 4 bytes)
$= \log_2(2^4)$

index = 4 bits (4 to 7)

tag = Rest (8 to 31)

(b) How many sets does this cache have? Explain.

16

(c) Below is a series of memory read references set to the cache from part (a). Assume that the cache is initially empty and classify each memory references as a hit or a miss. Identify each miss as either compulsory, conflict, or capacity. One example is shown. Hint: start by splitting the address into components. Show your work.

Address	Hit/Miss?	Miss Type?
0x300	Miss	Compulsory
0x1BC	Miss	Compulsory
0x206	"	"
0x109	"	"
0x308	"	Conflict
0x1A1	"	Compulsory
0x1B1	Hit	
0x2AE	Miss	compulsory
0x3B2	"	"
0x10C	Hit	
0x205	Miss	conflict
0x301	"	"
0x3AE	"	Compulsory
0x1A8	"	Conflict
0x3A1	Hit	
0x1BA	Hit	

(d) Calculate the miss rate and hit rate.

Hit rate $= 0.25 = \frac{4}{16}$

Miss rate $= 1 - 0.25 = 0.75$

4. **Cache organization :** Your company has an application that must be run as fast as possible. The hardware division of your company has come up with three separate first-level cache configurations:

Machine I:	Direct-mapped with one-word blocks
Machine II:	Direct-mapped with four-word blocks
Machine III:	Two-way set associative with four-word blocks

For these machines, the cache fill penalty is 4 cycles + 1 cycle for each word.
You did some experiments and measured the following instruction mix for the application:

$$Branch : 16\%, Load : 15\%, Store : 10\%, FloatInsts : 20\%, Integer : 39\%$$

Further, through a hardware cache monitor, you measured the following miss rates:

Machine I:	Instruction miss rate: 4%;	Data miss rate: 20%
Machine II:	Instruction miss rate: 2%;	Data miss rate: 16%
Machine III:	Instruction miss rate: 1.5%;	Data miss rate: 14%

Finally, the total CPI measured with Machine I is 3.0.

(a) Which of these machines spends the most time waiting for memory? Justify your answer:

M1: $5 * ((0.04 * 1) + (0.1 * 0.15) * 0.2) = 0.45$

M2: $8 * ((0.02 * 1) + (0.1 * 0.15) * 0.16) = \boxed{0.48}$

M3: $8 * ((0.015 * 1) + (0.1 * 0.15) * 0.14) = 0.40$

5 = full penalty for 1 word block
8 = " " " 4 word block

So M2

(b) Suppose we increase the associativity of Cache III (keeping the block-size constant). In this new configuration, the instruction miss rate goes to zero. Further, we measure a total CPI of 2.71. What is the data miss rate?

Base CPI without memory stalls

= 3.0 - 0.45 = 2,55

Thus, 2.71 - 2.55 = 0.16 of CPI waiting for memory.

So,

0.16 = 8 * (0.1 * 0.15) * Data miss rate

Data miss rate = 8 %.