COMP3211
Computer Architecture Tutorial 7

Jeremy Chan
jeremyc@cse.unsw.edu.au

Q01. [P7.7,7.8,7.20,7.21,7.22] Here is a series of address references given as word addresses:

```
1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17
```

a) Assuming a direct-mapped cache with 16 one-word blocks that is initially empty, label each reference in the list as a hit or miss and show the final contents of the cache.

Cache block	Access #1	Access #2	Access #3	Final Contents	Sequence	Hit/Miss
0				N/A	1	miss
1	1 (miss)	17 (miss)	17 (hit)	17	4	miss
2				N/A	8	miss
3	19(miss)			19	5	miss
4	4 (miss)	20 (miss)	4 (miss)	4	20	miss
5	5 (miss)	5 (hit)		5	17	miss
6	6 (miss)			6	19	miss
7				N/A	56	miss
8	8 (miss)	56 (miss)		56	9	miss
9	9 (miss)	9 (hit)		9	11	miss
10				N/A	4	miss
11	11 (miss)	43 (miss)		43	43	miss
12				N/A	5	hit
13				N/A	6	miss
14				N/A	9	hit
15				N/A	17	hit

b) Show the hits and misses and final cache contents for a direct-mapped cache with four-word blocks and a total size of 16 words.

In this example, we increase our line size to 4. When there is a miss for example for the first block, 4 words are read into the cache.

Cache block	word	Access #1	Access #2	Access #3	Access #4	Access #5	Access #6	Final Contents	Sequence	Hit/Miss
0	0	0(line)	16 (line)					16	1	miss
0	1	1 (miss)	17 (miss)		17(hit)			17	4	miss
0	2	2 (line)	18 (line)					18	8	miss
0	3	3(line)	19 (line)	19 (hit)				19	5	hit
1	4	4 (miss)		20(miss)	4 (miss)			4	20	miss
1	5	5 (line)	5 (hit)	21(line)	5 (line)	5(hit)		5	17	miss
1	6	6 (line)		22(line)	6 (line)		6 (hit)	6	19	hit
1	7	7(line)		23(line)	7 (line)			7	56	miss
2	8	8 (miss)	56 (miss)	8 (line)		40 (line)	8 (line)	8	9	miss
2	9	9(line)	57 (line)	9 (miss)		41(line)	9(miss)	9	11	hit
2	10	10(line)	58 (line)	10 (line)		42 (line)	10 (line)	10	4	miss
2	11	11(line)	59 (line)	11 (line)	11 (hit)	43(miss)	11 (line)	11	43	miss
3	12							N/A	5	hit
3	13							N/A	6	hit
3	14							N/A	9	miss
3	15							N/A	17	hit

c) Show the hits and misses and final cache contents for a two-way set-associative cache with one-word blocks and a total size of 16 words. Assume LRU Replacement.

Block	Set	Access #1	Access #2	Access #3	Final Contents	Sequence	Hit/Miss
0	0	8 (miss)			8	1	miss
1	0	56(miss)			56	4	miss
0	1	1 (miss)	9(miss)	9(hit)	9	8	miss
1	1	17(miss)	17(hit)		17	5	miss
0	2				N/A	20	miss
1	2				N/A	17	miss
0	3	19(miss)	43(miss)		43	19	miss
1	3	11(miss)			11	56	miss
0	4	4(miss)	4(hit)		4	9	miss
1	4	20 (miss)			20	11	miss
0	5	5(miss)	5(hit)		5	4	hit
1	5				N/A	43	miss
0	6	6(miss)			6	5	hit
1	6				N/A	6	miss
0	7				N/A	9	hit
1	7				N/A	17	hit

d) Show the hits and misses and final cache contents for a fully associative cache with one-word blocks and a total size of 16 words. Assume LRU Replacement.

Cache block	Access #1	Access #2	Final contents	Sequence	Hit/Miss
0	1 (miss)		1	1	miss
1	4 (miss)	4 (hit)	4	4	miss
2	8 (miss)		8	8	miss
3	5 (miss)	5 (hit)	5	5	miss
4	20 (miss)		20	20	miss
5	17 (miss)	17(hit)	17	17	miss
6	19 (miss)		19	19	miss
7	56 (miss)		56	56	miss
8	9 (miss)	9 (hit)	9	9	miss
9	11 (miss)		11	11	miss
10	43 (miss)		43	4	hit
11	6 (miss)		6	43	miss
12			N/A	5	hit
13			N/A	6	miss
14			N/A	9	hit
15			N/A	17	hit

e) Show the hits and misses and final cache contents for a fully associative cache with four-word blocks and a total size of 16 words. Assume LRU Replacement.

This is a fully set associative cache with line size of 4

Cache block	word	Access #1	Access #2	Access #3	Access #4	Access #5	Sequence	Hit/Miss	Cache block
0	0	0 (line)	16 (line)		40 (line)		1	miss	0
0	1	1 (miss)	17 (miss)		41 (line)		4	miss	1
0	2	2 (line)	18 (line)		42 (line)		8	miss	2
0	3	3 (line)	19 (line)	19 (hit)	43 (miss)		5	hit	1
1	4	4 (miss)		8 (line)			20	miss	3
1	5	5 (line)	5 (hit)	9 (miss)		9 (hit)	17	miss	0
1	6	6 (line)		10 (line)			19	hit	0
1	7	7 (line)		11 (line)	11 (hit)		56	miss	2
2	8	8 (miss)	56 (miss)	16 (line)			9	miss	1
2	9	9 (line)	57 (line)	17 (miss)			11	hit	1
2	10	10 (line)	58 (line)	18 (line)			4	miss	3
2	11	11 (line)	59 (line)	19 (line)			43	miss	0
3	12	20 (miss)	4 (miss)				5	hit	3
3	13	21 (line)	5 (line)	5 (hit)			6	hit	3
3	14	22 (line)	6 (line)		6 (hit)		9	hit	1
3	15	23 (line)	7 (line)				17	miss	2

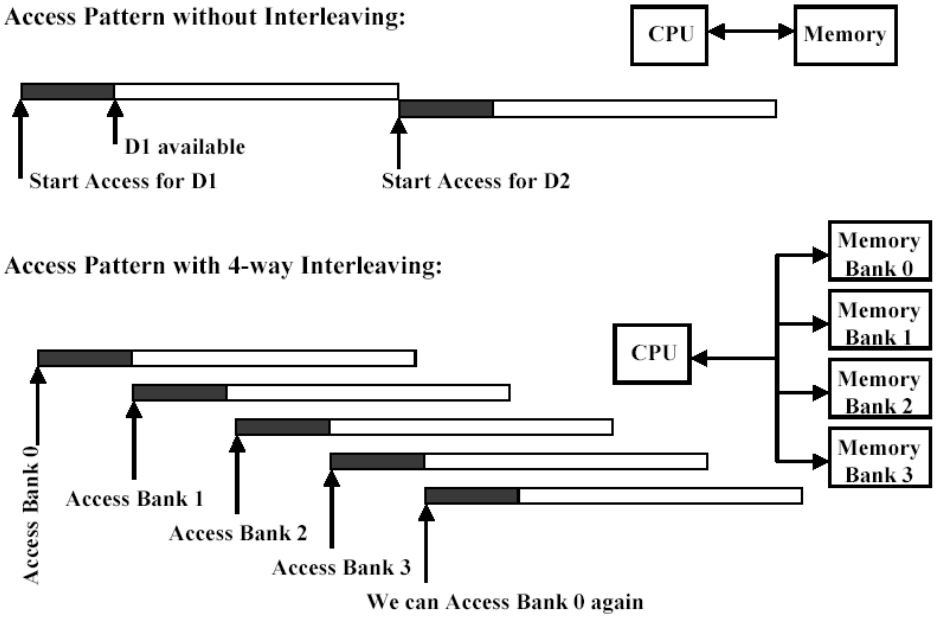
f) Which of the above cache organisations is best for the sequence of references given?

From the above, it can be seen that increasing the line size increases the hit rate due to the spatial locality of the input data set. In this case, increasing the associativity for line sizes of 1 also increases the hit rate for this input sequence. In b and e, are the best organisation both with line size of 4, there are 6 hits out of 16.

Q02. [P7.11] Consider a memory hierarchy using one of the three organisations for memory shown in the figure of Slide 28 in Lecture 10, http://www.cse.unsw.edu.au/~cs3211/coursework/L101_slides.pdf.

Assume that the cache block size is 16 words, that the width of the "Wide memory organisation" of the figure is four words, and that the number of banks in the "Interleaved memory organisation" is four. If the main memory latency for a new access is 10 cycles and the transfer time is 1 cycle, what are the miss penalties for each of these organisations?

Miss Penalty: Time to replace a block in the upper level + Time to deliver the block the processor



In this question, the cache line size is 16.

The time to deliver block to cache is given below:

Simple organisation = No. of Accesses * (Access Time + Reading Time)

Simple organisation = 16 x (10 + 1) = 176

Wide memory organisation = No. of Wide Accesses * (Access Time + Reading Time)
Wide memory organisation = 4 x (10 + 1) = 44

Interleaved organisation = No. of Interleaved Accesses * (Access Time + No. of Banks * Reading Time per bank)
Interleaved organisation = 4 x (10 + 4 x 1) = 56

Q03. [P7.12] Suppose a processor with a 16-word block size has an effective miss rate per instruction of 0.5%. Assume that the CPI without cache misses is 1.2. Using the memories described in the figure of Q02, how much faster is the processor when using the wide memory than when using narrow or interleaved memories?

0.005 x 176 + 1 x 1.2 = 2.08

0.005 x 44 + 1 x 1.2 = 1.42

2.08/1.42 = 1.46

46 % faster

Q04. [Based on P7.38] If all misses are classified into one of three categories - compulsory, capacity, or conflict - which misses are likely to be reduced when a program is rewritten so as to require less memory? How about if the clock rate of the machine that the program is running on is increased? How about if the main data structure used is changed from an array to a tree structure in order to aid searching?

If the program is rewritten to require less memory, it could reduce all three types of misses. Compulsory miss are reduced because there is now fewer compulsory misses (fewer instructions). Capacity miss may also be reduced because the program is now smaller and may fit in cache. Conflict

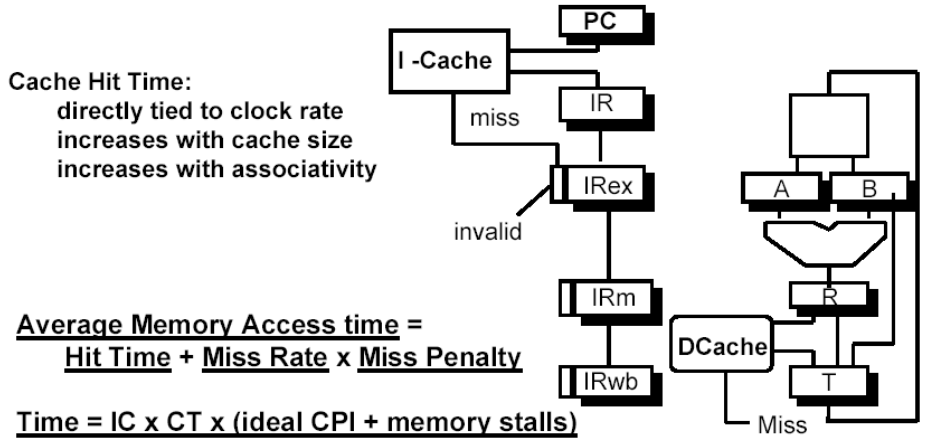
misses can go both ways. Code segments may also be smaller and cause less conflict but depend on placement and type of cache.

Clock frequency does not change the number of misses and the types of misses.

If the main data structure is changed from an array to a tree, the memory would likely be more fragmented. This is likely to exhibit less spatial locality. A tree structure is less efficiently stored when compared to an array hence, it will take up more space in memory. Depending on the access patterns, it is likely to exhibit higher compulsory, capacity and conflict misses.

Q05. With reference to Slide 27 of Lecture 11, http://www.cse.unsw.edu.au/~cs3211/coursework/L11_slides.pdf, which depicts a five stage pipelined datapath, what factors should be considered in the design of the I- and D-cache? What would style and size of cache would you recommend?

State any assumptions you have made.



I-Cache Design
The design of the I-Cache will be dependent on application parameters such the number of active instructions (typically the loops that are run without conflicting). Your loops should fit in cache to avoid capacity misses.

The clockable frequency of the pipeline helps determine the type of I-cache you use. The clockable frequency of the cache decreases with cache size and associativity. Your I-cache should match the target frequency of the processor

The type of memory (what burst modes it supports) may affect your selection of line size of your instruction caches.

D-Cache
The design of data cache will depend on the size of the dataset and how they are accessed (access patterns).