

CSCI 516: Fundamental Concepts in Computing and Machine Organization

Homework Assignment 7

Due on 12/06/2022 11:55PM

Requirement

- Show the **steps of calculation**, instead of giving the final result only.
- Finish your assignment **individually**
- Submit **.pdf** file which contains your solutions to the D2L.

1. Problem 5.5 from Textbook.

5.5 For a direct-mapped cache design with a 64-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
63–10	9–5	4–0

5.5.1 [5] <\$5.3> What is the cache block size (in words)?

5.5.2 [5] <\$5.3> How many blocks does the cache have?

5.5.3 [5] <\$5.3> What is the ratio between total bits required for such a cache implementation over the data storage bits?

Beginning from power on, the following byte-addressed cache references are recorded.

Address												
Hex	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884
Dec	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.5.4 [20] <\$5.3> For each reference, list (1) its tag, index, and offset, (2) whether it is a hit or a miss, and (3) which bytes were replaced (if any).

5.5.5 [5] <\$5.3> What is the hit ratio?

5.5.6 [5] <\$5.3> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>. For example,

<0, 3, Mem[0xC00]-Mem[0xC1F]>

5.5.1) Block Size can be determined by examining the offset bits. Since the problem gives 4 offset bits (4-0), we can calculate the Cache block size in words by finding the value of 2^5 or 32 words per cache block.

5.5.2) The number of blocks in a Cache can be determined by examining the index bits. Since the problem gives an index of 9-5 the 5 bits can be used in the expression 2^5 which allows one to determine that there are 32 blocks in a directly mapped Cache.

5.5.3) The ratio between total bits required for such a cache implementation over the data storage bits can be determined by looking at the Valid bits for each line of Cache. Since the problem gives a tag of 63-10, 54 bits and 32 words each of a size "x", are required for the tag field.

$$\frac{\text{Total bits}}{\text{Total data bits}} = \frac{(54 + 32(x))}{(32(x))} =$$

5.5.4) First I need to convert each binary to its 16 bit binary form.
To check whether it is a hit or a miss we must examine the index values.
I have highlighted them for clarity.
The tag section is to the left of the highlighted index values.

00 0000 0000 0000 0000

A mandatory first miss occurred

04 0000 0000 0000 0100

Hit the index and tag are the same
as 0x00

10 0000 0000 0001 0000

Hit the index and tag are the same
as 0x04

84 0000 0000 1000 0100

This is a mandatory miss

E8 0000 0000 1110 1000

This is a mandatory miss

A0 0000 0000 1010 0000

This is a mandatory miss

400 0000 0100 0000 0000

This is a miss the index is the same as 0x04 but the tags are different

1E 0000 0000 0001 1110

This is a miss the index is the same as 0x400 but the tags are different

8C 0000 0000 1000 1100

This is a hit the index is the same as the 4th Frame

C1C 0000 1100 0001 1100

This is a miss the index is the same as 0x1E but the tags are different

B4 0000 0000 1011 0100

This is a hit the index is the same as 0xA0

884 0000 1000 1000 0100

This is a miss Since the frame needs replacement

5.5.5) The hit ratio is $4/12$ or 0.33

5.5.6) $\langle 4, 2, \text{Mem}[0 \times 880] - \text{Mem}[0 \times 89F] \rangle$
 $\langle 0, 3, \text{Mem}[0 \times C00] - \text{Mem}[0 \times C1F] \rangle$
 $\langle 5, 0, \text{Mem}[0 \times A0] - \text{Mem}[0 \times BF] \rangle$
 $\langle 7, 0, \text{Mem}[0 \times E0] - \text{Mem}[0 \times FF] \rangle$

2. Problem 5.10 from Textbook.

5.10 In this exercise, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional to capacity. Assume that main memory accesses take 70 ns and that 36% of all instructions access data memory. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

	L1 Size	L1 Miss Rate	L1 Hit Time
P1	2 KiB	8.0%	0.66 ns
P2	4 KiB	6.0%	0.90 ns

5.10.1 [5] <§5.4> Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

5.10.2 [10] <§5.4> What is the Average Memory Access Time for P1 and P2 (in cycles)?

5.10.3 [5] <§5.4> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster? (When we say a “base CPI of 1.0”, we mean that instructions complete in one cycle, unless either the instruction access or the data access causes a cache miss.)

For the next three problems, we will consider the addition of an L2 cache to P1 (to presumably make up for its limited L1 cache capacity). Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

L2 Size	L2 Miss Rate	L2 Hit Time
1 MiB	95%	5.62 ns

5.10.4 [10] <§5.4> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

5.10.5 [5] <§5.4> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

5.10.6 [10] <§5.4> What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P1 without an L2 cache?

5.10.7 [15] <§5.4> What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P2 without an L2 cache?

5.10.1)

$$P_1 \text{ Clock Rate} = 1/0.66 = 1.515 \text{ GHz}$$

$$P_2 \text{ clock Rate} = 1/0.9 = 1.11 \text{ GHz}$$

5.10.2)

P_1 Average Memory Access Time

$$= .66 + (.08 \times 70) = 6.26 \text{ ns}$$

$$= L_1 \text{ hit time} + (L_1 \text{ miss rate} \times \text{Memory Access time})$$

P_2 Average Memory Access Time

$$= .90 + (.06 \times 70) = 5.1 \text{ ns}$$

5.10.3) P_2 is faster since it has a lower total CPI

P_1 Total CPI

$$= 1 + [(70 \times .08)/.66] \times .36 = 4.054$$

$$= \text{Base CPI} + [(\text{Memory Access Time} \times L_1 \text{ miss rate})/L_1 \text{ Hit time}] \times \frac{\text{No. of instructions}}{\text{instructions}}$$

P_2 Total CPI

$$= 1 + [(70 \times .06)/.90] \times .36 = 2.68$$

5.10.4) Average memory access time for P_1

with the addition of a L2 Cache

$$= L_1 \text{ hit time} + L_1 \text{ Miss rate} \times (L_2 \text{ Hit time} + L_2 \text{ Miss Rate} \times \text{MAT})$$

$$= .66 + .08 \times (5.62 + .95 \times 70)$$

$$= .66 + .08 \times (5.62 + 66.5)$$

$$= 6.4296 \text{ ns}$$

Since the average memory access time increased, the AMAT performs worse after adding an extra L2 Cache.

5.10.5)

P_1 w/ L2 Cache \rightarrow Total CPI

$$\text{Total CPI} = \text{Base CPI} + \text{Number of memory instructions} \times$$

$$[L_1 \text{ miss rate} \times (L_2 \text{ hit in cycles} + L_2 \text{ memory miss in cycles})]$$

5.10.7)

For P_1 to be faster than P_2

$$.66 \times (1 + 0.36 \times X \times 106) < P_2 \text{ time}$$

$$P_2 \text{ time} = 2.68 \times .9 = 2.41 \text{ ns/instruction}$$

$$X \leq 69.29\% \text{ Miss rate}$$

The miss rate should be less than 69.29%

5.10.6)

For P_1 to be faster than P_1 ,

$$1 + .08 \times (9 + x \times 107) < 9.56$$

$$x < 91.59\%$$

The miss rate should be less than 91.59%