Name(Pinyin):
Email(Prefix):

Computer Architecture Homework 6 Solution

2019 Spring Apr.22

Instructions:

Homework 6 is due on May. 6, covers the content of caches and float-points, please refer to the lecture slices. You can print it out and write on it, and scan it into a pdf, or you can take photos or write Latex if you want, just remember: you must create a **PDF** and upload to the **Gradescope**, please assign the questions properly on Gradescope, otherwise you will lose 25% of points.

Tell us your feeling after finish it. Thank you!

Question Set 1. Direct Mapped Cache

[30 points] In a 16-bit byte-addresses machine, the clock frequency is 3GHz. We have a cache with properties as follows:

1. Cache size is 64 Bytes;
2. Block size is 4 Bytes;
3. Cache hit time is 2 cycles;
4. Cache miss penalty is 100 cycles;

1-A. What the width of each field of following address bit assignment:

TAG:10	Set index:4	Block offset:2

Give me the progress how you think about it.
Answer 6pt + Analysis 4pt

1-B. We will access the data of addresses as follows. Fill in the blanks. It is about the index, tag (in decimal) and whether there is a hit or miss. If there is a miss, then give what type is the miss (either compulsory or replace).

Addresses (serially access)	Tag/Index	Hit, Compulsory or Replace
0x0000	0/0	Compulsory
0x0004	0/1	Compulsory
0x0008	0/2	Compulsory
0x000c	0/3	Compulsory
0x1000	64/0	Replace
0x1004	64/1	Replace
0x1008	64/2	Replace
0x100c	64/3	Replace
0x0000	0/0	Replace
0x0004	0/1	Replace

1-C. Calculations. (Show progress, worth 50% pts)
1-C-i: missing rate: (4 pt.)
1
1-C-ii: AMAT (ns): (3 pt.)
34
1-C-iii: AMAT if we don't have this cache (ns): (3 pt.)
33.3
AMAT w/ cache = hit time + miss rate * miss time
clock frequency = 3GHz = 3·10^9 Hz
period = 1/(3·10^9) = 0.333 ns
hit time = 2 cycles => 0.667 ns
miss time = 100 cycles => 33.3 ns
from problem 1.3, miss rate = 1
so AMAT w/ cache = 0.667ns + (1*33.3ns) = 33.967 ns = 34 ns
AMAT w/o cache = miss time = 33.3 ns

Question Set 2. Two-Way Set Associative Cache

From QS 1. We change the block size to 8 Bytes. And implemented two-way set associative cache. The parameters are shown as follows:

1. Cache size is 64 Bytes;
2. 16-bit byte-addresses machine;
3. Block size is 2 words;

2-A. What the width of each field of following address bit assignment:

TAG: 11	Set index: 2	Block offset: 3

Give me the progress how you think about it.
Answer 6pt + Analysis 4pt

2-B. We will access the data of addresses as follows. Fill in the blanks. Fill in the blanks. It is about the index, tag (in decimal) and whether there is a hit or miss. If there is a miss, then give what type is the miss (either compulsory or replace). (Each blank worth 1 pt.)

Addresses (serially access)	Tag/Index	Hit, Compulsory or Replace
0x0000	0/0	Compulsory
0x0004	0/0	Hit
0x0008	0/1	Compulsory
0x000c	0/1	Hit
0x1000	128/0	Compulsory
0x1004	128/0	Hit
0x0000	0/0	Hit
0x0100	8/0	Replace
0x0000	0/0	Replace (Hit for LRU)
0x1004	128/0	Replace

2-C. Calculations.
2-C-i. Missing rate: (5 pt.)
0.667. (0.5 for LRU.)
2-C-ii. Assume the new cache miss time is 200 cycles and hit time is 3 cycles. Calculate the AMAT in ns. Round to the nearest tenth. (5 pt.)
The miss penalty stays the same because we still go to memory (200 cycles) exactly as we did in problem 2.
The hit time increases slightly because determining a hit now requires muxing between the different elements of the cache set. The miss penalty stays the same as in problem 2 since two-way set associativity does not affect the penalty incurred on a miss.
AMAT = hit time + miss rate * miss time
= (3 cycles + 0.667 * 200 cycles) * 0.333 ns/cycle = 19.8 ns
= (3 cycles + 0.5 * 200 cycles) * 0.333 ns/cycle = 34.3 ns (for LRU)

Question Set 3. Floating Point Numbers

We consider the IEEE 32-bit floating point representation except with a 7 bit exponent (bias of 63) and a denorm implicit exponent of -62.

4-A. Convert `-95.2` to that form. In hexadecimal.
0xc57ccccc

4-B. Convert `0x4a23a000` into floating point, specify infinities as `+inf` and `-inf`, and not a number as `NaN`.
2333(decimal)

4-C. What is the smallest non-infinite positive integer it CANNOT represent? (integer is `XX.0000`). Tell me why.
（1.1111…）b * 2^63 = (1 + 1 - 2^(-24)) * 2^63=2^63 - 2^39.

4-D. What's the smallest positive value it can represent that is not a denorm? Leave your answer as a power of 2. Tell me why.

Exponent field: 0b0000001 So exponent is 1 - 63 = -62 => 2^(-62)
4-E. What's the smallest positive value it can represent? Leave your answer as a power of 2. Tell me why.
2^-62*2^-24 = 2^-86