CS 61C Fall 2018 — Caches II, Floating Point

Discussion 6: October 1, 2018

1 Code Analysis

Given the follow chunk of code, analyze the hit rate given that we have a byte-addressed computer with a total memory of **1 MiB**. It also features a **16 KiB** Direct-Mapped cache with **1 KiB** blocks.

```
#define NUM_INTS 8192 // 2^13
int A[NUM_INTS]; // A lives at 0x10000
int i, total = 0;
for (i = 0; i < NUM_INTS; i += 128) {
    A[i] = i; // Line 1
}
for (i = 0; i < NUM_INTS; i += 128) {
    total += A[i]; // Line 2
}
```

1.1 How many bits make up a memory address on this computer?

$1\,MiB = 2^{20}B \rightarrow \log 2^{20} = 20$ bits T: 20 − 10 − 4 = 6 bits

1.2 What is the T:I:O breakdown?

O: $1\,KiB = 2^{10}B \rightarrow \log 2^{10} = 10$ bits I: $\frac{16\,KiB}{1\,KiB} = 16$ indices $\rightarrow \log 16 = 4$ bits

1.3 Calculate the cache hit rate for the line marked Line 1:

jumping 4·128 = 512 B each loop → 2 elems fit in each block ⇒ A[i] pulls in A[i+128] ⇒ 50% HR

1.4 Calculate the cache hit rate for the line marked Line 2:

array size = $2^{13} \cdot 4 = 2^{15}$ B → twice as big as cache :(→ cannot fit inside → 50% HR for reason above.

2 AMAT

Recall that AMAT stands for Average Memory Access Time. The main formula for it is:

AMAT = Hit Time + Miss Rate * Miss Penalty

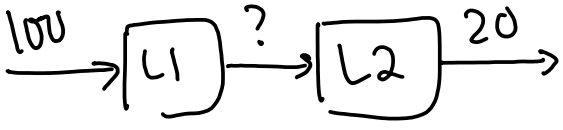
We also have two types of miss rates, global and local. Global is calculated as: Fraction of ALL accesses that missed at that level over all accesses total. Whereas local is calculated: Fraction of ALL access that missed at that level over all access to that level total.

Note L_1 global = local

$G = \frac{\text{Misses Here}}{\text{Total Access}}$

$L = \frac{\text{Misses Here}}{\text{Accesses Here}}$

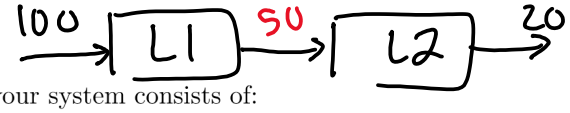
2.1 An L2$, out of 100 total accesses to the cache system, missed 20 times. What is the global miss rate of L2$?



$\frac{20}{100} = 20\%$

L1 miss rate
$100 \cdot MR = 100 \cdot 50\% = 50$

2.2 If L1$ had a miss rate of 50%, what is the local miss rate of L2$?



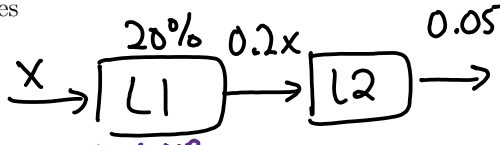
$\frac{20}{50} = 40\%$

Suppose your system consists of:

1. An L1$ that hits in 2 cycles and has a local miss rate of 20%
2. An L2$ that hits in 15 cycles and has a global miss rate of 5%
3. Main memory hits in 100 cycles

notice Local MR = $\frac{\text{Global MR}}{\text{L1 MR}}$

2.3 What is the local miss rate of L2$?



$LMR = \frac{0.05x}{.2x} = \frac{.05}{.2} = .25 = 25\%$

2.4 What is the AMAT of the system?

L1 GMR L2 GMR

2 ways ① using GMR: $2 + 20\% \cdot 15 + 5\% \cdot 100 = 2 + 3 + 5 = 10$ cycles

L1 Hit time L2 hit time main mem ht

② using LMR

$2 + 20\%\left(15 + 25\% \cdot 100\right)$

L1 LMR L2 LMR

2.5 Suppose we want to reduce the AMAT of the system to 8 cycles or lower by adding in a L3$. If the L3$ has a local miss rate of 30%, what is the largest hit time that the L3$ can have?

$2 + 20\%\left(15 + 25\% \cdot \left(H + 30\% \cdot 100\right)\right) \leq 8$

$H = 30$ cycles

3 Floating Point

The IEEE 754 standard defines a binary representation for floating point values using three fields:

- The *sign* determines the sign of the number (0 for positive, 1 for negative)
- The *exponent* is in **biased notation** with a bias of 127
- The *significand or mantissa* is akin to unsigned, but used to store a fraction instead of an integer

The below table shows the bit breakdown for the single precision (32-bit) representation.

1	8	23
Sign	Exponent	Mantissa/Significand/Fraction

For normalized floats:

Value $= (-1)^{Sign} * 2^{Exp-Bias} * 1.\textbf{significand}_2$

For denormalized floats:

Value $= (-1)^{Sign} * 2^{Exp-Bias+1} * 0.\textbf{significand}_2$

Exponent	Significand	Meaning
0	Anything	Denorm
1-254	Anything	Normal
255	0	Infinity
255	Nonzero	NaN

Write this on your cheat sheet!

forgetting denorm or ∞ on a test question... feels bad man.

3.1 How many zeroes can be represented using a float?

100... 0 → −0
000... 0 → +0

2

3.2 What is the largest finite positive value that can be stored using a single precision float?

3.3 What is the smallest positive value that can be stored using a single precision float?

3.4 What is the smallest positive normalized value that can be stored using a single precision float?

3.5 Cover the following numbers from binary to decimal or from decimal to binary:

- 0x00000000
- 8.25
- 0x00000F00
- 39.5625
- 0xFF94BEEF
- $-\infty$

4 Extra Stuff on Caches!

4.1 Heres some practice involving a 2-way set associative cache. This time we have an 8-bit address space, 8 B blocks, and a cache size of 32 B. Classify each of the following accesses as a cache hit (H), cache miss (M) or cache miss with replacement (R). For any misses, list out which type of miss it is.

Address	T/I/O	Hit, Miss, Replace
0b0000 0100	0 0 4	M comp
0b0000 0101	0 0 5	H
0b0110 1000	6 1 0	M comp
0b1100 1000	12 1 0	M comp
0b0110 1000	6 1 0	H
0b1101 1101	13 1 5	R comp
0b0100 0101	4 0 5	M comp
0b0000 0100	0 0 4	H
0b1100 1000	12 1 0	R, capacity

4.2 What is the hit rate of our above accesses?

always draw your cache!