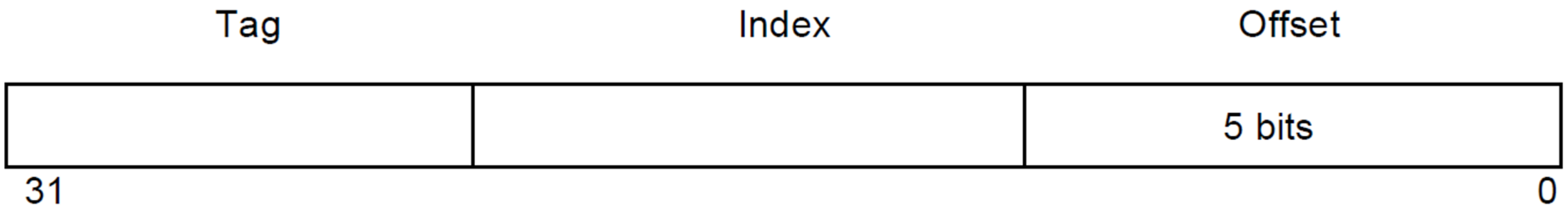
review

1

Exercise

- Consider a 32-bit physical memory space and a 32 KiB 2-way associative cache with LRU replacement.

You are told the cache uses 5 bits for the offset field. Write in the number of bits in the tag and index fields in the figure below.



Exercise

Tag	Index	Offset
18 bits	**9 bits**	5 bits

31 … 0

- For the same cache, after the execution of the following code:

```
int ARRAY_SIZE = 64 * 1024;
int arr[ARRAY_SIZE]; // *arr is aligned to a cache block
/* loop 1 */ for (int i = 0; i < ARRAY_SIZE; i += 8) arr[i] = i;
/* loop 2 */ for (int i = ARRAY_SIZE - 8; i >= 0; i -= 8)
                  arr[i+1] = arr[i];
```

- 1. What is the hit rate of loop 1? What types of misses (of the 3 Cs), if any, occur as a result of loop 1?
- 2. What is the hit rate of loop 2? What types of misses (of the 3 Cs), if any, occur as a result of loop 2?

```
int ARRAY_SIZE = 64 * 1024;
int arr[ARRAY_SIZE]; // *arr is aligned to a cache block
/* loop 1 */ for (int i = 0; i < ARRAY_SIZE; i += 8) arr[i] = i;
/* loop 2 */ for (int i = ARRAY_SIZE - 8; i >= 0; i -= 8)
                  arr[i+1] = arr[i];
```

- 1. What is the hit rate of loop 1? What types of misses (of the 3 Cs), if any, occur as a result of loop 1? 0, Compulsory Misses
- 2. What is the hit rate of loop 2? What types of misses (of the 3 Cs), if any, occur as a result of loop 2? 9/16, Capacity Misses

2. Miss rate

Local miss rate – the fraction of references to one level of a cache that miss

Local Miss rate L2\$ = \$L2 Misses / L1\$ Misses

Global miss rate – the fraction of references that miss in all levels of a multilevel cache

- L2\$ local miss rate >> than the global miss rate

Global Miss rate = L2\$ Misses / Total Accesses

= (L2\$ Misses / L1\$ Misses) × (L1\$ Misses / Total Accesses)

= Local Miss rate L2\$ × Local Miss rate L1\$

Example: 1000 references, 40 misses in L1 cache and 20 misses in L2
Calculate L1 and L2's local and global miss rate

3. AMAT

AMAT = hit time + miss rate × miss penalty

Example: 1000 references, 40 misses in L1 cache and 20 misses in L2
Local miss rates: 4% (L1), 50% (L2) = 20/40
Global miss rates: 4% (L1), 2% (L2)

1. Suppose that you have a cache system with the following properties. What is the AMAT?
   a) L1$ hits in 1 cycle (local miss rate 25%)
   b) L2$ hits in 10 cycles (local miss rate 40%)
   c) L3$ hits in 50 cycles (global miss rate 6%)
   d) Main memory hits in 100 cycles (always hits)

4. Floating point

- IEEE 754
- 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* is akin to unsigned, but used to store a fraction instead of an integer.

32bits

Sign	Exponent	Significand
1 bit	8 bits	23 bits

64bits

uses 11 bits for the exponent (and thus a bias of 1023) and 52 bits for the significand.

For normalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \times 2^{(\text{Exponent} - \text{Bias})} \times 1.\text{significand}_2$$

For denormalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \times 2^{(\text{Exponent} - \text{Bias} + 1)} \times 0.\text{significand}_2$$

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

Exercises

1. How many zeroes can be represented using a float?

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

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

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

5. Convert the following numbers from binary to decimal or from decimal to binary:

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

Answer

1. How many zeroes can be represented using a float? 2

2. What is the largest finite positive value that can be stored using a single precision float?
   0x7F7FFFFF = $(2 - 2^{-23}) \times 2^{127}$

3. What is the smallest positive value that can be stored using a single precision float?
   0x00000001 = $2^{-23} \times 2^{-126}$

4. What is the smallest positive normalized value that can be stored using a single precision float?
   0x00800000 = 2^{-126}

5. Convert the following numbers from binary to decimal or from decimal to binary:

   0x00000000   8.25   0x00000F00   39.5625   0xFF94BEEF   -∞

0x00000000 = 0
8.25 = 0x41040000
0x000000F0 = $(2^{-12} + 2^{-13} + 2^{-14} + 2^{-15}) \times 2^{-126}$
39.5625 = 0x421E4000
0xFF94BEEF = NaN
-∞ = 0xFF800000