1 Pre-Check

This section is designed as a conceptual check for you to determine if you conceptually understand and have any misconceptions about this topic. Please answer true/false to the following questions, and include an explanation:

1.1 For the same cache size and block size, a 4-way set associative cache will have fewer index bits than a direct-mapped cache.

True.

1.2 Any cache miss that occurs when the cache is full is a capacity miss.

False   *compulsory*

conflict

1.3 Increasing cache size by adding more blocks always improves (increases) hit rate.

False

program   characterizing

2   Understanding T/I/O

When working with caches, we have to be able to break down the memory addresses we work with to understand where they fit into our caches. There are three fields:

Tag - Used to distinguish different blocks that use the same index. Number of bits: (# of bits in memory address) - Index Bits - Offset Bits

Index - The set that this piece of memory will be placed in. Number of bits: $\log_2$(# of indices)

Offset - The location of the byte in the block. Number of bits: $\log_2$(size of block)

Given these definitions, the following is true:

$\log_2$(memory size) = address bit-width = # tag bits + # index bits + # offset bits

Another useful equality to remember is:

$$\text{cache size} = \text{block size} * \text{num blocks}$$

2.1 Assume we have a direct-mapped byte-addressed cache with capacity 32B and block size of 8B. Of the 32 bits in each address, which bits do we use to find the index of the cache to use?   *3-4 bits*

4 blocks
27 bit 2 bit 3 bit

2.2 Which bits are our tag bits? What about our offset?   *31-5 bits   2-0 bits*

2.3 Classify each of the following byte memory accesses as a cache hit (H), cache miss (M), or cache miss with replacement (R). Tip: Drawing out the cache can help you see the replacements more clearly.

Address	T/I/O	Hit, Miss, Replace
0x00000004	0 0 4	M *compulsory*
0x00000005	0 0 5	H
0x00000068	3 1 0	M *compulsory*
0x000000C8	6 1 0	R *compulsory*
0x00000068	3 1 0	R *conflict*
0x000000DD	6 3 5	M *compulsory*
0x00000045	2 0 5	R *compulsory*
0x00000004	6 0 4	R *capacity*
0x000000C8	6 1 0	R *capacity*

3 Cache Associativity

In the previous problem, we had a Direct-Mapped cache, in which blocks map to specifically one slot in our cache. This is good for quick replacement and finding out block, but not good for efficiency of space!

This is where we bring associativity into the matter. We define associativity as the number of slots a block can potentially map to in our cache. Thus, a Fully-Associative cache has the most associativity, meaning every block can go anywhere in the cache.

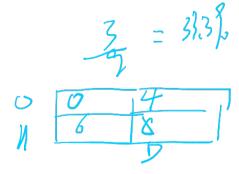For an N-way associative cache, the following is true:

$$N * \# \text{ sets} = \# \text{ blocks}$$

3.1 Here's 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. Assume that we have an LRU replacement policy (in general, this is not the case).

Address	T/I/O			Hit, Miss, Replace	
0b0000 0100	0	0	4	M	
0b0000 0101	0	0	5	H	
0b0110 1000	6	1	0	M	
0b1100 1000	C	1	0	M	X
0b0110 1000	6	1	0	H	X
0b1101 1101	D	1	5	R	
0b0100 0101	4	0	5	M	
0b0000 0100	0	0	4	H	
0b1100 1000	C	1	0	R	

(margin notes)
#block = 4
set = 2.

4 bit 1 bit 3 bit.

compulsory
compulsory
compulsory
compulsory.
compulsory.
capacity

3.2 What is the hit rate of our above accesses?

$$\frac{3}{9} = 33.3\%$$

4   The 3 C's of Cache Misses

4.1 Go back to questions 2 and 3 and classify each M and R as one of the 3 types of misses described below:

1. Compulsory: First time you ask the cache for a certain block. A miss that must occur when you first bring in a block. Reduce compulsory misses by having longer cache lines (bigger blocks), which bring in the surrounding addresses along with our requested data. Can also pre-fetch blocks beforehand using a hardware prefetcher (a special circuit that tries to guess the next few blocks that you will want).

2. Conflict: Occurs if, hypothetically, you went through the ENTIRE string of accesses with a fully associative cache (with an LRU replacement policy) and wouldn't have missed for that specific access. Increasing the associativity or improving the replacement policy would remove the miss.

3. Capacity: Capacity misses are independent of the associativity of your cache. If you hypothetically ran the ENTIRE string of memory accesses with a fully associative cache (with an LRU replacement policy) of the same size as your cache, and it was a miss for that specific access, then this miss is a capacity miss. The only way to remove the miss is to increase the cache capacity.

Note: The test you can use to see if a miss is a conflict miss is the same as the test you can use to see if a miss is a capacity miss.

Note: There are many different ways of fixing misses. The name of the miss doesn't necessarily tell us the best way to reduce the number of misses.

5 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. Assume that your cache begins cold.

```
#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
}
```

(handwritten annotations: memory cache Performance; #block = 16; write / read; 128, $2^7 \cdot 2^2 = 2^9$; total may be left in cache; total may be in a register optimized by compiler)

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

2^{20} 20 bits

5.2 What is the T:I:O breakdown?

6 bits 4 bits 10 bits

5.3 Calculate the cache hit rate for the line marked Line 1: $\dfrac{16}{16 \times 2} = 0.5$

0

5.4 Calculate the cache hit rate for the line marked Line 2:   0.5

(handwritten work)

$$3) \quad S = 2^{10} \quad S+2^9 \qquad 0 \quad m \cdot H$$
$$S+2^{10} \quad S+2^{10}+2^9 \qquad 1$$
$$S+2^{20} \quad S+2^{20}+2^8 \qquad \vdots$$

$$\frac{2^{12}}{2^4} = 2^8$$

15	m	H
16	m	H
17	m	H

2^{12} ... $S+2^{40}$

$S+2$