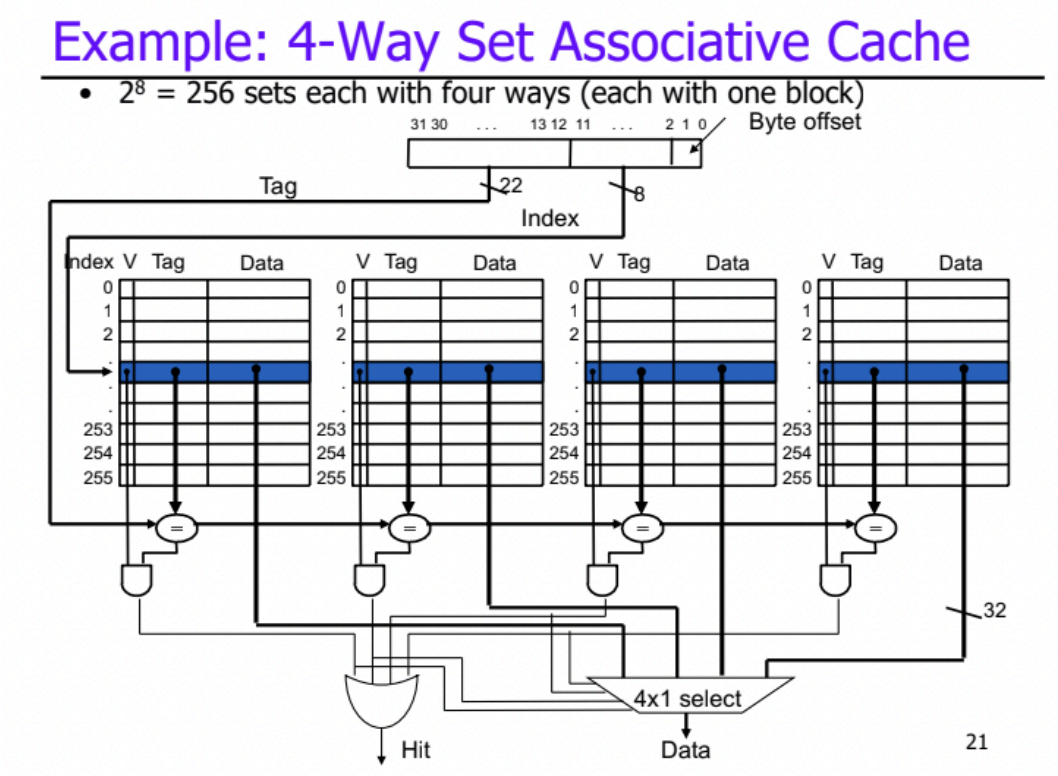
Lecture 11 Cache 2 Exercises

Q1:
Assume:
- A processor has a 64KB 2-way set associative cache
- The cache access uses physical addresses only
- A physical address is 48 bits long
- Each block holds 64 bytes of data
- Tag overhead includes the valid bit and tag bits

How much is the tag overhead in percent?

Solution:
The total number of cache blocks = 64KB / 64B = 1024
Because 2-way set associative, the index bits in the address only need to identify each of the 512 (=1024 / 2) cache sets
Number of bits in index = log2(1024 / 2) = 9
Number of bits in byte offset = log2 (64) = 6
Number of bits in tag = 48 – 9 – 6 = 33
Each cache block has a tag and a valid bit, i.e., with a 2-way set associative cache, each cache set has two tags and valid bits for each way, respectively



Therefore, tag overhead = (33+1)/8/(64) * 100% = 6.6%

Q2:
DAXPY is an acronym for a common double precision scientific loop of the form: Y[i] = A*X[i] + Y[i] where X and Y are arrays. Assume that X and Y are both 64 MB and our total cache capacity is 512 MB, and each array is stored sequentially in memory. We run the following loop:

```
extern double *X;
  extern double *Y;
  start_time = current_time();
  for (outer = 0; outer < 100000; outer++){
    for (i=0; i< 8 * 1024 * 1024; i++){
         Y[i] = A*X[i] + Y[i];
    }
  }
  end_time = current_time();
  duration = end_time - start_time;
```

When we analyze the performance counters, we find we are taking many last-level cache (LLC) misses. What can be done to the design to minimize the execution time of this code?

A. Increase the associativity of the LLC
B. Decrease the LLC line size
C. Decrease the associativity of the LLC
D. None of the above

Answer: A
Cache capacity 512MB > 64MB + 64MB, so it is not capacity miss → many misses can be conflict misses
So, increasing the associativity can help