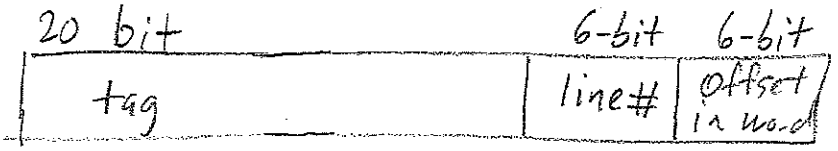
Comp. Arch. HW #6

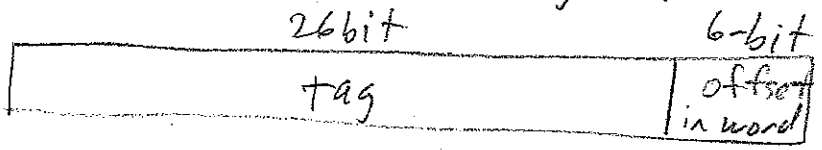
4.11 32-bit addr., byte addressable - Memory
Cache - 64byte line size ($2^6 = 64$)

(a)



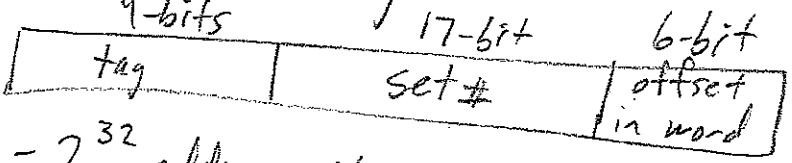
- number of addressable units - # bytes in memory 2^{32}
- # blocks in main memory = $\frac{2^{32}}{2^6} = 2^{26}$ blocks
- # cache lines $2^6 = 64$
- 20 bits is size of tag (given?)

(b)



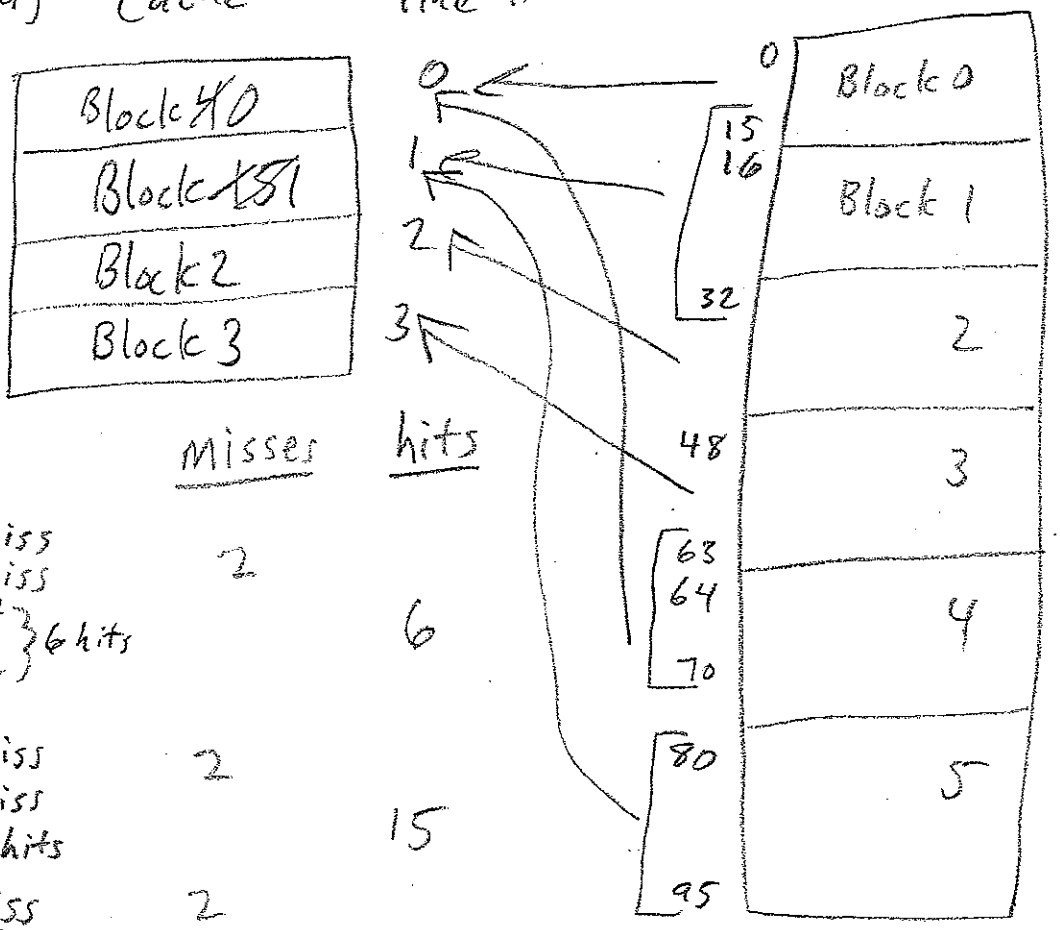
- 2^{32} addr. units
- 2^{26} blocks in main memory
- # cache lines - Cannot tell
- 26-bits in tag

(c)



- 2^{32} addr. units
- 2^{26} blocks in main memory
- 4 lines in each set
- 2^{17} sets in cache
- $2^2 \times 2^{17} = 2^{19}$ lines in cache
- 9-bit tags

4.18 (a) Cache line #



Trace	Misses	hits
63 - miss 64 - miss 65 - hit ⋮ 70 - hit (6 hits)	2	6
1st iteration 15 - miss 16 - miss 17 ⋮ 31 (15 hits) 32 - miss 80 - miss 81 ⋮ 95 (15 hits)	2 2	15 15
2nd iteration 15 - hit 16 - miss 17 ⋮ 31 32 (16 hits) 80 - miss 81 ⋮ 95 (15 hits)	1 1	17 15

(0.5)

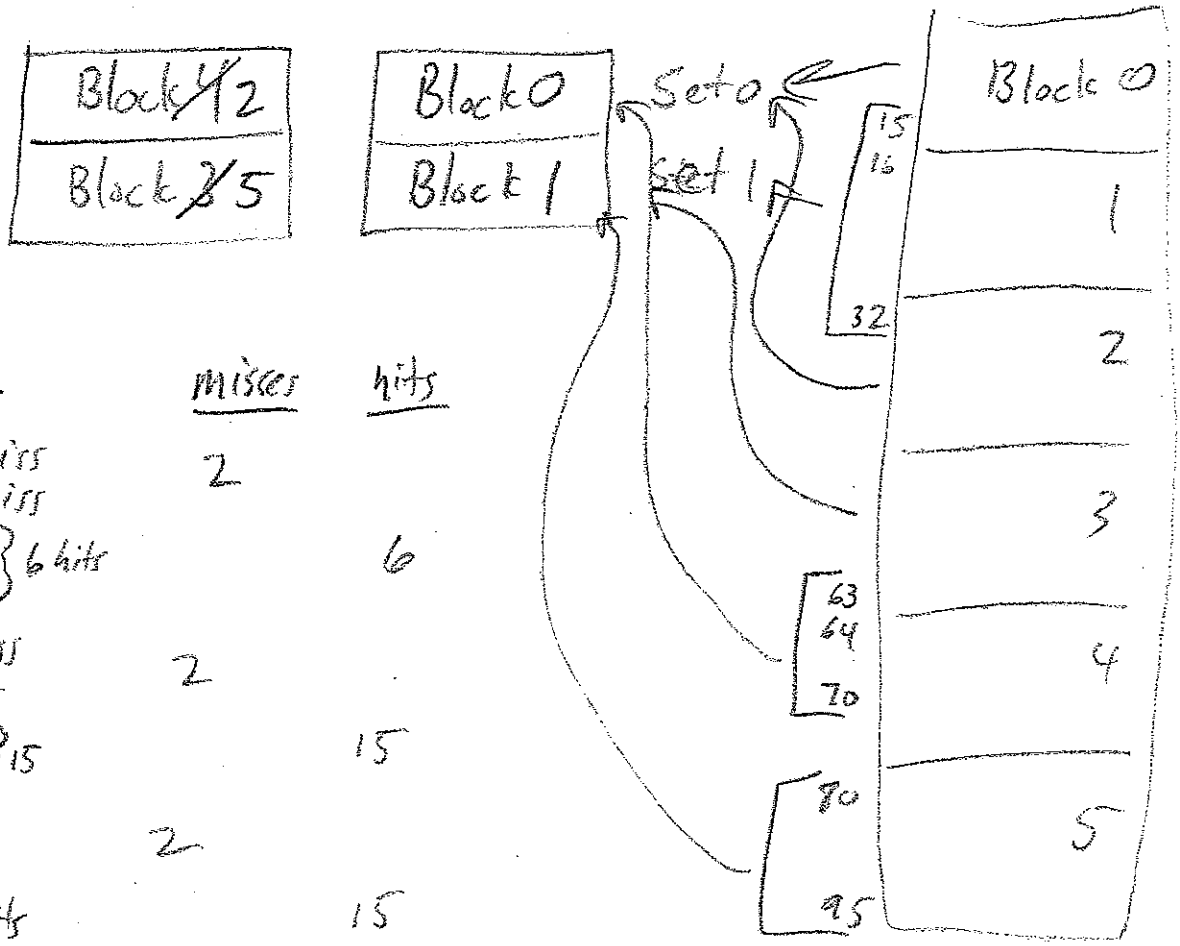
3rd - 10th same as 2nd

$$\frac{2 \times 8}{24} + \frac{32 \times 8}{324} = 348 \text{ total accesses}$$

$$\text{hit ratio} = \frac{324}{348} \approx 0.931$$

HW6-2

4.18(b)



Trace		misses	hits
1st iteration (0.5)	63 – miss 64 – miss 65 – hit 70 – hit	2	6
	15 – miss 16 – miss 17 – hit 31 – hit	2	15
	32 – miss 80 – miss 81 95	2	15
2nd	15 – hit 16 – hit 31 – hit 32 – hit 80 – hit 95 – hit		34

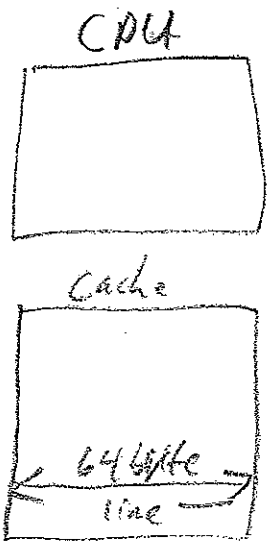
3rd – 10th same as 2nd iteration

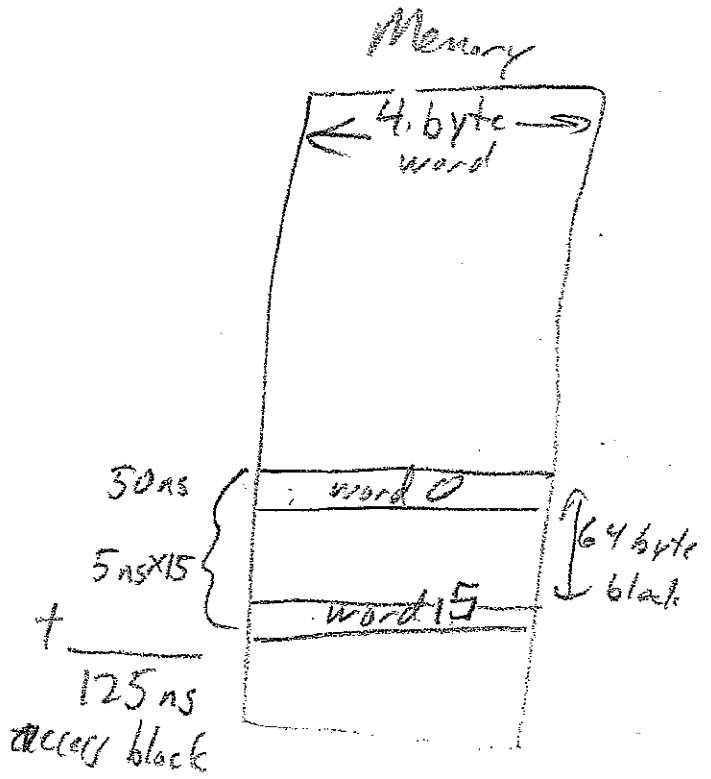
$\frac{34 \times 8}{6} + 342 = 348$ total accesses

hit ratio $= \frac{342}{348} = 0.983$

4.21

(a)





H=0,95

Access time on cache miss = 2,5ns + 125ns + 2,5ns = 130ns

(discover miss) (access block) (reexecute for hit)

1

Average memory access time = (hit time) x (hit ratio) + (miss time) x (miss ratio)

= 2,5ns x 0,95 + 130ns x 0,05

= 2,375ns + 6,5ns = 8,875ns

(b) line size of 128, H of 0,97.

Access time on cache miss = 2,5ns + 50ns + 5ns x 31 + 2,5ns = 210ns

Average memory access time = 2,5 x 0,97 + 210 x 0,03 =

2,425 + 6,3 = 8,725ns

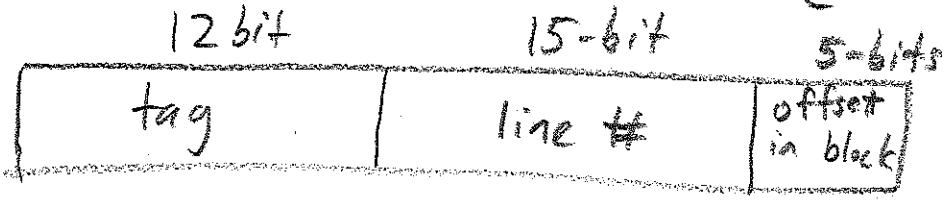
5 Yes, the bigger line size reduces average memory access time

Question A:

a) $\#\text{ cache lines} = \frac{\text{cache size}}{\text{block size}} = \frac{2^{20}}{2^5} = 2^{15}$ lines

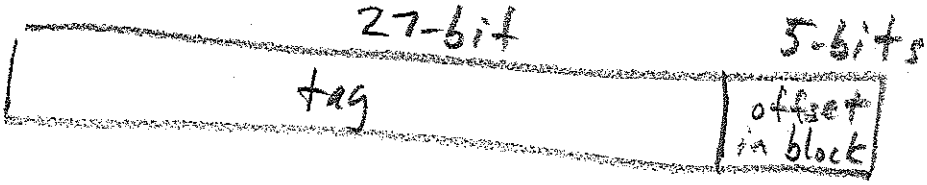
b) one - each memory block can be stored in only one cache line

c)



d) 2^{15} since an memory block can map to any cache line

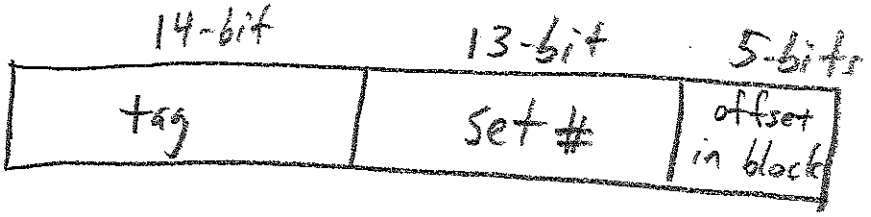
e)



f) 4

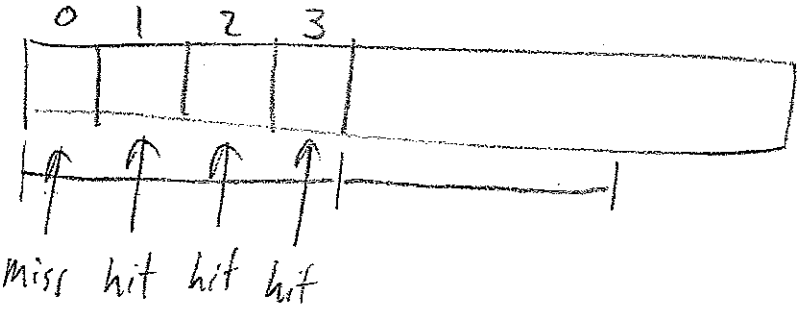
g) $\frac{2^{15}\text{ lines}}{2^2} = 2^{13}$ sets of size 4

h)



Question B

Code A accesses M by scanning down each row. Each row is split into memory blocks, ie.,



The first element from the block will cause a miss, but the others will cause a hit.

0,5

Code B accesses M by scanning down each column. A block read into the cache due to a miss for the first element will probably be forced from the cache before other elements in that block can be accessed.

Question C. Assume an 6-disk RAID array. Each disk has 100 MB/sec data transfer rate.

a) Fill in the following table to compare the following RAID levels **assuming no disk failures.**

RAID Level	Data transfer rate for single, large I/O request	Number of concurrent, small READ requests	Number of concurrent, small WRITE requests
0 (large strip)	100 MB/sec	6	6
0 (bit-wise interleaving)	600 MB/sec	1	1
1	100 MB/sec 300 MB/sec	6 1	3 1
2 (3 data, 3 hamming)	300 MB/sec	1	1
3	500 MB/sec	1	1
4	100 MB/sec	5	1
5	100 MB/sec	6	3

large block bit-wise

b) Fill in the following table to compare the following RAID levels **assuming one disk failure.**

RAID Level	Data transfer rate for single, large I/O request	Number of concurrent, small READ requests	Number of concurrent, small WRITE requests
0 (large strip)	~~100 MB/sec~~ crashed	not available	—
0 (bit-wise interleaving)	~~600 MB/sec~~ crashed	not available	—
1	100 MB/sec 300 MB/sec	5 1	3 1
2	300 MB/sec	1	1
3	500 MB/sec	1	1
4	100 MB/sec	4 or 5	1
5	100 MB/sec	5	3

blocked bit-wise