Homework 4 (Due Jan. 24)

1. Here is a series of memory address references given as word address: 1, 1, 4, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17, 32, 4, 22. Assuming a direct-mapped cache with 16 one-word blocks that is initially empty, label each memory reference in the list as a hit or miss and show the final contents of the cache (valid bit, tag and data) by filling the following two tables. Assume memory address is 6-bit long. What is the miss rate of the cache? tag — %16

Reference	Hit or miss
1	Miss
1	HIT
4	Miss
4	HIT
8	Miss
5	Miss
20	Miss
17	Miss
19	Miss
56	Miss
9	Miss
11	Miss
4	HIT
43	Miss
5	HIT
6	Miss
9	HIT
17	HIT
32	Miss
4	HIT
22	Miss

Cache Block	Valid bit	Tag	Data
0	0	10	32
1	0,1	00,01	1, 17
2	0		
3	0	01	19
4	0,1	00, 01, 00	4, 20, 4
5	0,1	00	5
6	0	11, 00, 01	56, 6, 22
7	0		
8	0	00	8
9	0	00	9
10	0		
11	0	00, 10	11, 43
12	0		
13	0		
14	0		
15	0		

Miss rate = ___15/21___ ← tag/reference

2. Find the AMAT (average memory access time) in ns for a machine with 2GHZ clock, a miss penalty of 35 clock cycles, a 85% hit rate, and a cache access time (i.e hit time) of 2 clock cycle.

$$AMAT = \frac{(2 + \overset{100-85=15}{15\%} \cdot 35)}{(2 \cdot 10^9)} = 3.625\text{ ns}$$

3. A memory has 32-bit addresses. It also has a 256 KB cache. Blocks are 64 bytes in size. Note 1K = 2^{10}, 1M = 2^{20}

a. How many blocks are there in the cache if the cache is a direct-mapped cache?

Cache / blocks

256 / 64 = 4 K blocks

b. How many sets are there in the cache if the cache is a 2-way set-associative cache?

2 K blocks

c. Show the partitioning of a memory address into the appropriate fields, if the cache is a direct-mapped cache, and 2-way set associative, respectively. That is to find how many bits needed for tag, block/set index and byte offset, respectively.

Block offset :

$64B = 2^{n?} = 2^6 = 6$ bits

Block index :

$4K = 2^{n?} \cdot 2^{10}$

$= 2^2 \cdot 2^{10}$

$= 2^{12} = 12$ bits

tag :

32 - 6 - 12

= 14 bits

d. What is the total size of the cache if the cache is a direct-mapped cache?

# of blocks · (valid bit + tag + data)

$= 2^{12} \cdot (1 + 14 + 64(8))$

$= \frac{2{,}158{,}592}{8}$ bits

= 269 824 B

e. What is the total size of the cache if the cache is a 2-way set associative cache?

Block offset

$64B = 2^6$

= 6 bits

Block index

2 sets

$= 2 \quad 10$

$= 2^{11} = 11$ bits

tag

32 - offset - index

32 - 6 - 11

= 15 bits

# of blocks · (valid bit + tag + data)

$2^{12} \cdot (1 + 15 + 64(8))$

$= \frac{2{,}162{,}688}{8}$ bits

= 270336 B

4. For a direct-mapped design with a 64-bit address, the following bits of the address are used to access the cache.

Tag	**Index**	**offset**
63-10	9-5	4-0
54 bits	5 bits	5 bits

a. What the cache line size (in words)?

block offset

$= 5$ bits

$= 2^5$

$= \frac{32 \text{ bytes}}{4} = 8$ words

b. How many entries does the cache have?

Block index

$= 5$ bits

$= 2^5$

$= 32$ entries

c. What is the ratio between total bits required for such a cache implementation over the data storage bits?

$$\frac{\text{total bits}}{\text{data bits}} = \frac{(\text{valid bit} + \text{tag} + \text{data}) \quad (1 + 54 + 32(8))}{(32(8))} = 1.22$$

5. Using the series of references given in Question #1, show the hits and misses and final cache contents for **a two-way** set associative cache with **one word block** and a total size of 16 words. Assume **LRU** replacement. What is the hit rate of the cache?

LRU → Least Recently used

1 word block w/ 16 words

each set has size of 2 ...

instead of % 16
use %8

tag	index	Reference	Hit or miss
000	001	1	Miss
000	001	1	Hit
000	100	4	Miss
000	100	4	Hit
001	000	8	Miss
000	101	5	Miss
010	100	20	Miss
010	001	17	Miss
010	011	19	Miss
111	000	56	Miss
001	001	9	Miss
001	011	11	Miss
000	100	4	Hit
101	011	43	Miss
000	101	5	Hit
000	110	6	Miss
001	001	9	Hit
010	001	17	Hit
100	000	32	Miss
000	100	4	Hit
010	110	22	Miss

7 hit
21 reference

Set #	Block #	Valid bit	Tag	Data
0	0	0, 1	001, 100	8, 32
	1	0, 1	111	56
1	0	0, 1	000, 001	1, 9
	1	0, 1	010	17
2	0	0		
	1	0		
3	0	0, 1	010, 011	19, 43
	1	0, 1	001	11
4	0	0, 1	000	4
	1	0, 1	010	20
5	0	0, 1	000	5
	1	0		
6	0	0, 1	000	6
	1	0, 1	010	22
7	0	0		
	1	0		

Hit rate = 7/21

hit/references

6. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 64-bit integer. Considering the various of i, j, A[i][j], B[i][0], A[j][i],

```
for (i=0; i<8; i++)
   for (j=0; j<8000; j++)
         A[i][j] = B[i][0] + A[j][i]);
```

getting changed

changes constantly

(a) How many 64-bit integers can be stored in a 16-byte cache block?

$$\frac{(16 \cdot 8)}{64} = 2$$

(b) References to which variables exhibit temporal locality? (instructions in loops)

B[i][0] and i, j

(c) References to which variables exhibit spatial locality? (array data)

A[i][j]

7. A hard drive has ~~three~~ 3 platters, with 512 tracks per surface. Each track has 256 sectors, with 512 data bytes per sector. It spins at 7200 rpm. The average seek time is 8 ms. The data transfer rate is 10MB/sec, and the controller overhead is 2ms.

a. What is the drive capacity?

Platters x surfaces x tracks per surface x track sectors x data(Bytes)

$3 \cdot 2 \cdot 512 \cdot 256 \cdot 512$

$= 402,653,184$ B

b. What is the average access time for the drive? 8MS according to question

$$\text{Seek time} + \frac{1/2}{(\text{rpm} / 60\text{s} \cdot 10^3)}$$

10^3 to convert s to MS

MS rotational latency:

$$\frac{1/2}{\left(\frac{7200\text{rpm}}{60 \cdot 10^3}\right)} = 4.17\text{ms}$$

databytes per sector

MS transfer time

data transfer rate

$$\frac{512\text{ B}}{10\text{ MB/s}}$$

$= 51.2\text{s} \cdot 10^3$

$= .0512$ms

controller delay

2 ms

from question

seek time + rotational latency + transfer time + controller delay

8 + 4.17 + .0512 + 2

= 14.2212 ms

8. There are several parameters that impact the overall size of the page table of a virtual memory. Below are several key page table parameters: Virtual address size = 32 bits, each page size = 8kB, and page table entry size = 4 bytes. Given the above parameters, calculate the total page table size for a system with 2GB physical RAM running 6 applications. pg 97 slides begin

32-bit virtal address

8 KB pages $= \log_2(8 \cdot 2^{10}) = 13$ bit page offset

$= 2^3 \cdot 2^{10} = 2^{13} = 13$ bit page offset

One word page table entry (4 bytes)

Virtual page number (vpn)

virtual address size − page offset

32 − 13 = 19 bits

Number of entries

applications $\cdot (2^{vpn} \cdot \text{entry})$

$6(2^{19} \cdot 4)$

= 12582912 B

= 12 MB