Computer Architecture

Fall, 2022

Week 14

2021.09.26

組別：_____ 簽名：_____

_____

[Group10]

Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of ~~32~~ 8-bit memory address references, given as word addresses.

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

For each of these references, identify the binary word address, the tag, the index, and the offset given a direct-mapped cache with two-word blocks and a total size of eight blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

8 block → 3 index

2 word/block → 1 offset

	tag 4bits	index 3bits	block offset 1bits	
3 → 0000 0011	0000	001	1	M
180 → 1011 0100	1011	010	0	M
43 → 0010 1011	0010	101	1	M
2 → 0000 0010	0000	001	0	H
191 → 1011 1111	1011	111	1	M
88 → 0101 1000	0101	100	0	M
190 → 1011 1110	1011	111	0	H
14 → 0000 1110	0000	111	0	M
181 → 1011 0101	1011	010	1	H
44 → 0010 1100	0010	110	0	M
186 → 1011 1010	1011	101	0	M
253 → 1111 1101	1111	110	1	M

[Group11]

Given the following code snippets, select the one that has better performance and explain why.

1. Suppose the 2d-array x is in row major order (x[i] [j] and x[i] [j+1] is adjacent)

(a)

```
for (i =0; i <5000; i=i+1){
    for (j =0; j<100; j=j+1){
        x[i][j] = 2*x[i][j];
    }
}
```

(a) row major order
the row would be loaded into page
then accessing same row gives higher performance
∋ spacital locality

(b)

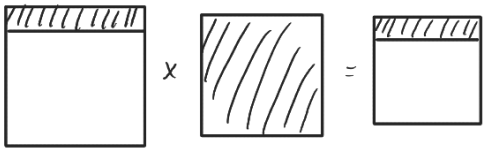
```
for (j = 0; j < 100; j= j+1){
    for (i=0; i<5000; i=i+1){
        x[i][j] = 2*x[i][j];
    }
}
```

2. NxN Matrix Multiplication (N=9)

(a)

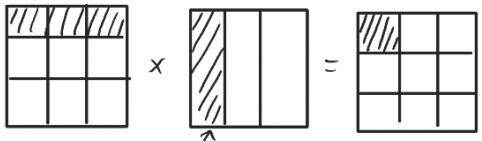
```
for (i = 0; i< N; i=i+1){
    for (j =0; j < N; j = j+1){
        r = 0;
        for (k = 0; k<N; k = k +1){
            r = r + y[i][k] * z[k][j];
        }
        x[i][j] = r;
    }
}
```

(a)



太大 cache 可能存不了會 flush 掉

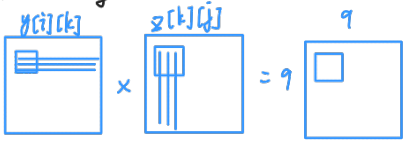
(b)



可重覆利用

temporal locality

√ (b)




```
for (jj = 0; jj < N; jj = jj+3){
    for (kk = 0; kk < N; kk = kk+3){
        for (i = 0; i<N; i= i+1){
            for (j = jj; j<min(jj+3,N); j= j+1){
                r = 0;
                for (k = kk; k <min(kk+3,N); k=k +1){
```

```
                    r = r + y[i][k] * z[k][j];
                }
                x[i][j] = x[i][j] + r;
            }
        }
    }
}
```

[Group7]

Please answer the following questions using the caches with the following property respectively:

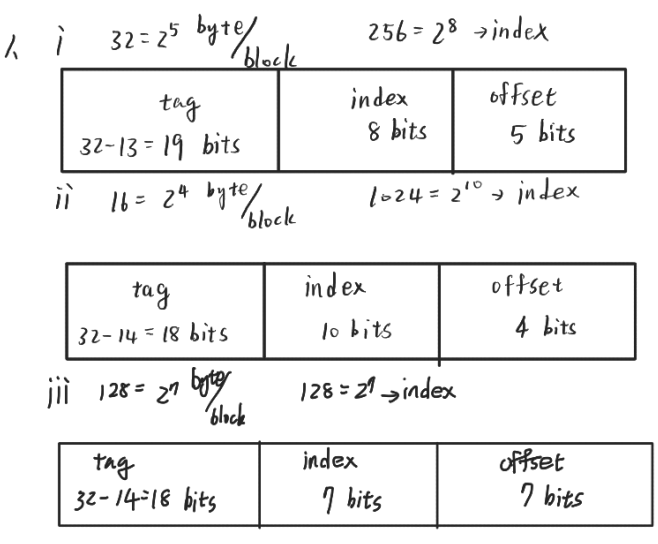
i. 32 bytes per block, 256 blocks in a cache

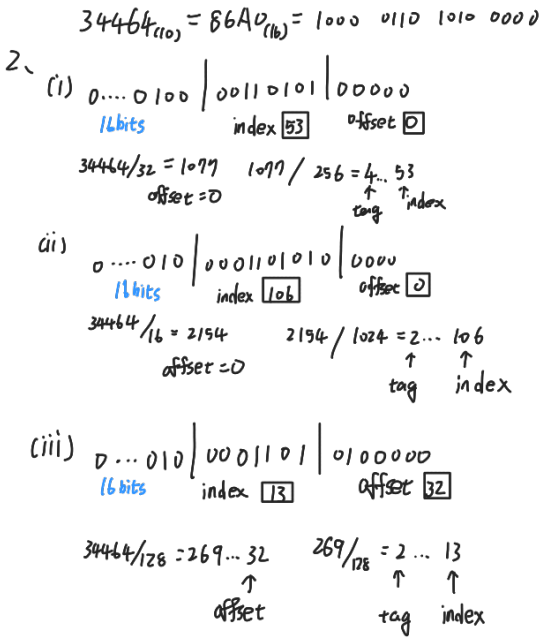
ii. 16 bytes per block, 1024 blocks in a cache

iii. 128 bytes per block, 128 blocks in a cache

1. Subdivide the memory address for a 32-bit word into tag, index, and offset.

2. What does $34464_{(10)} = 86A0_{(16)}$ map with the above caches?





[Group3]

✓ (A) DRAM is slow but cheap and dense; SRAM is fast but expensive, not very dense.

~~(B)~~ Cache is the component with the highest level in the memory hierarchy. register

✓ (C) The main reason we need cache is the performance gap between Memory and CPU

✓ (D) SRAM is faster than DRAM , and both of their access time for all locations are the same (random access)

✓ (E) If cache has 64 blocks and each blocks have 16 bytes/block. The cache block number which addresses 1200 map is 001011_2 $\frac{1200}{16}=75$ offset=0 $75/64=1\cdots 11$ (tag, index)

✓ (F) SRAM and DRAM are both volatile memories.

~~(G)~~ For upper level, DRAM is a better choice than SRAM because upper level needs to be ~~faster~~ slow and ~~smaller~~ big.

~~(H)~~ In direct map cache, if we access to a location in cache with valid bit = 1, there is no cache miss. 有 cache miss

[Group5]

1.Please choose the correct answer. (If it is wrong, please provide reasons.)

~~(A)~~ Although DRAM needs to be refreshed to prevent data missing, it's cheaper and ~~faster~~ slower than SRAM.

✓ (B)Loop is an example of Temporal locality.

~~(C)~~The unit of swapping data between Cache and Memory is Blocks, and is managed by the ~~OS~~ hardware.

~~(D)~~we need to save the full address ~~of tag~~ in Cache. 只要存 high order bit

✓(E)write buffer is FIFO.

✓ (F)the larger block size of cache, the larger miss penalty.

[Group 6]

~~a.~~ In a fixed-sized cache, the larger the block size is, the lower the miss rate will be, because of the Spatial Locality. 不一定

✓ b. We can apply some techniques of hash to do the Block Placement.

~~c.~~ Using direct mapping, to know which particular block is stored in a cache location, we also need to store the whole memory address of the block in the Tag file as well as data.

~~d.~~ Using direct mapping, for a cache with 32-bits memory address、2K words、1 word per block , there are 20-bits Tag and 10-bits Index in the cache table.

$2048 = 2^{11}$

2^2 byte/block

	tag	index	offset
d.	19 bits	11 bits	2 bits

a. result fewer blocks

[Group14]

Assuming a cache of 64 (2^6) blocks with 16 (2^4) bytes per block, to which block numbers do the following addresses map?

a. 1728

b. 876

c. 4

offset = 4 byte/block index = 6 bits

tag = 22 bits

a. $1728_{(10)} = 0110\ 1100\ 0000$

0 ... 0 01 | 1011 00 | 0000

22bits

tag = 1
index = 44
offset = 0

1728/16 = 108 ... 0 ↑offset

108/64 = 1 ... 44

↑ tag ↑ index

b. $876_{(10)} = 110\ 110\ 1100$

0 ... 00 | 11011 0 | 1100

22bits

tag = 0
index = 54
offset = 12

876/16 = 54 ... 12 ↑offset

54%64 = 0 ... 54

↑ tag ↑ index

c. $4_{(10)} = 0100$

0 ... 0 | 0 ... 0 | 0100

22bits 6bits 4bits

tag = 0
index = 0
offset = 4

4/16 = 0 ... 4

0 % 16 = 0