体系结构 Homework 2   3190102196   展翼飞

(a) block offset : Block size = 32 Bytes = 2^5 block offset = 5 bits

index : number of sets = $\frac{\text{Cache Size}}{\text{Block Size x Associativity}}$ = $\frac{8kB}{32B \times 2}$ = 128 sets = 2

index field = 7 bits

Tag : 38 - 7 - 5 = 26 bits

So

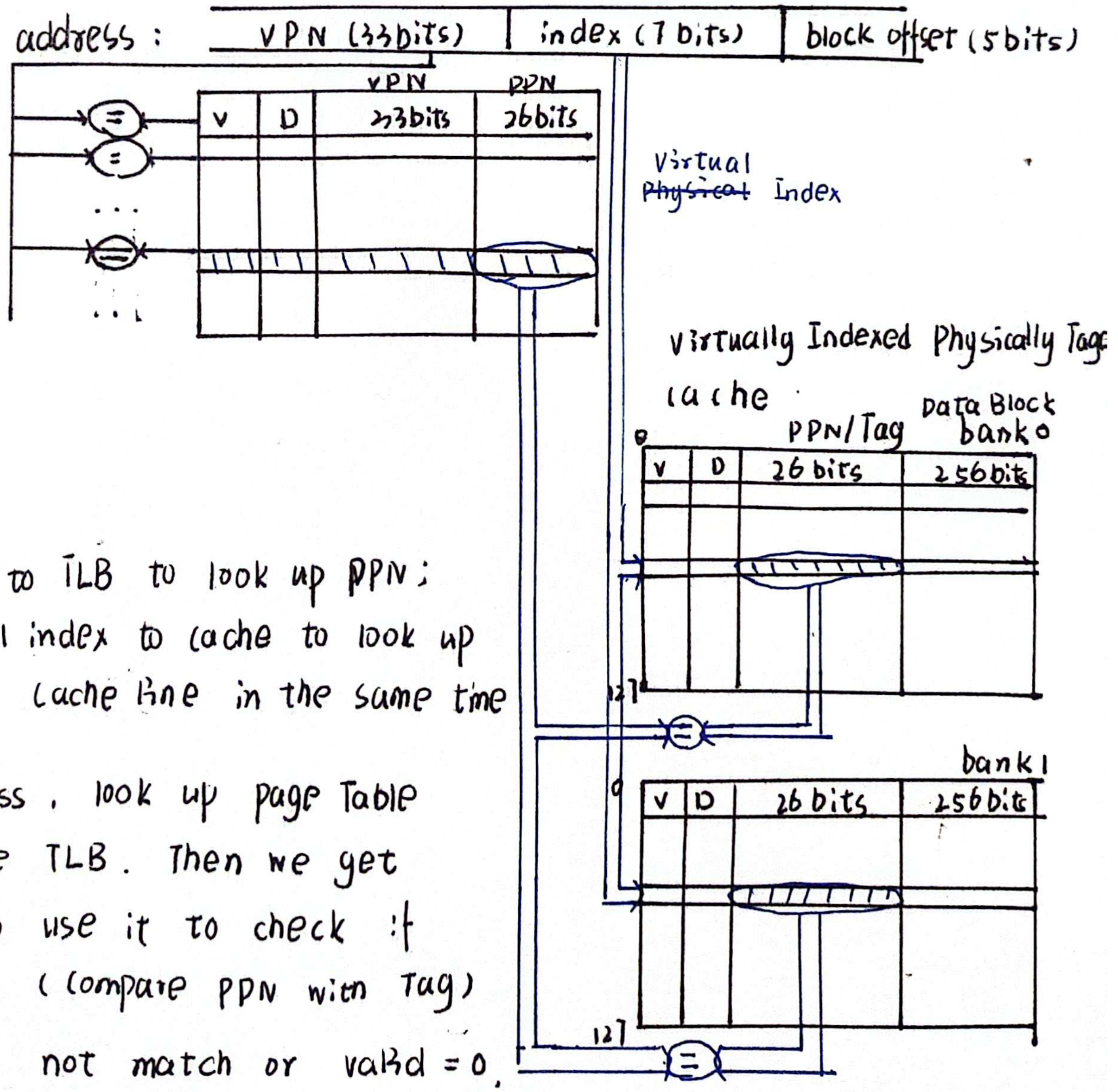
Tag	Index	Block offset
26 bits	7 bits	5 bits

(b)

valid bit	Dirty bit	Tag (26 bits)	Data (256 bits)

(c) virtual address :

VPN (33bits)	index (7 bits)	block offset (5bits)



(d)

① send VPN to TLB to look up PPN;
send virtual index to cache to look up accorrding cache line in the same time

② if TLB miss, look up page Table and update TLB. Then we get PPN, then use it to check if cache hit (compare PPN with Tag)

③ If tag do not match or valid = 0, retrive it from next level memory, then we get the data Block
If hit, get data directly.

Question 2:

AMAT = Hit time + Miss Rate × Miss Penalty

for Machine A:

$AMAT_a = 8 + 0.08 \times 50 = 12ns$

for machine B: → L_2 Hit Time + Miss Rate × Miss Penalty

$AMAT_b = 2 + [0.15 \times (20 + 0.10 \times 50)] = 5.75\ ns$

So B will have better performance

Question 3: 1.0 GHz = 1ns per cycle

(a) L_2 access time = 12 ns if ignore L_2 reaction time ~~(access time)~~

L_2 to L_1 bus transfer time $= \frac{32\ Bytes}{16\ Bytes} \times \frac{1}{266MHZ} = 7.52ns$

then L_1 Miss Penalty $= 12 + 7.52 = 19.52\ ns$

L_2 Miss penalty $= 80ns + \frac{64}{16} \times \frac{1}{133MHZ} = 30.08ns + 80ns = 110.08ns$

AMAT for Instruction access $AMAT_I = \cancel{8}\,1 + (0.02 \times (19.52ns + 0.15 \times 110.08ns))$

$AMAT_I = 1.72\ ns$

(b) L_2 to L_1 D-Cache bus transfer time $= \frac{16\ Bytes}{16\ Bytes} \times \frac{1}{266MHZ} = 3.76ns$

$L_{1D}\ mp = 12 + 3.76 = 15.76\ ns$

$L_2\ mp = 80ns + \frac{64}{16} \times \frac{1}{133MHZ} = 110.08\ ns$

$AMAT_{D\text{-}read} = \cancel{8}\,1 + (0.05 \times (15.76 + 0.15 \times 110.08)) = 2.62\ ns$

(c) $AMAT_{D\text{-}write}$ = L_1 Hit Time + 0.1 × (D-cache Miss Rate × Miss Penalty)

$= \cancel{8}\,1 + 0.1 \times (0.05 \times (15.76 + 0.15 \times 110.08)) = 1.16\ ns$

(d) CPI = Base CPI* + I frequency × $AMAT_I$ + L frequency × $AMAT_{D\text{-}read}$

+ S frequency × $AMAT_{D\text{-}write}$

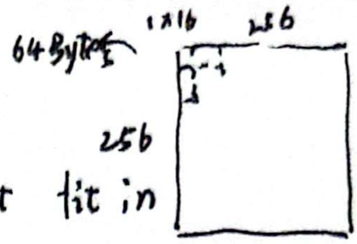
$= 1.35ns + 0.7 \times 1.72ns + 0.2 \times 2.62ns + 0.1 \times 1.16ns$

$= 3.19\ ns$

Question 4 :

(a) single precision number = 4 Bytes

When B = 16 a submatrix Line could just fit in a block



To take advantages of blocked execution, the L_1 cache should at least fit in 2 submatrix (read and write)

that is 16 + 16 = 32 Blocks = 2^{11} Bytes = 2KB

(b)

blocked version : each submatrix Line only need to fetch 1 time

then misses = $2 \times 16 \times (\frac{256}{16})^2 = 2^{13}$ times

= 8192 times

unblocked version :

Inner Loop : 1 miss for read and 16 misses for write every 16 num

then misses = $17 \times (\frac{256}{16}) \times 256$ = 69632 times

(c)
```
for(int i=0 ; i<256 ; i+=b){
    for (int j=0 ; j<256 ; j+=b){
        // Transpose submatrix
        for (ii = i ; ii < min(i+b, 256) ; ii++){
            for (jj = j ; jj < min(j+b, 256) ; jj++){
                output [jj][ii] = input [ii][jj]
            }
        }
    }
}
```

(d)

Question 5.

cache size 在一定时，block 越大，一个 block 装载的数据越多，越能利用空间局部性 miss rate 下降，然而过大时，能够存储的 blocks 较 一次 miss penalty 也更大

Question 6:

1.(1) 2、(1) 3.(2) 4.(3) 5.(2) 6.(4) 7.(2)

8.(4) 9.(2) 10.(2) 11、(4) 12、(4) ~~13.(~~