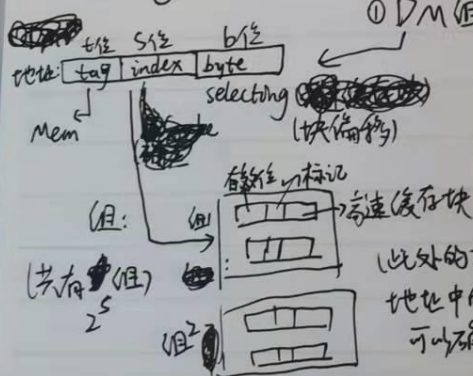


再读

Cache  $\neq$  Buffer

1. Principle: locality (局部性)
2. Mapping: 利用mem中数据更新 cache, 如何寻址?
3. Organization

2. { PM: Direct Mapping  
FA: Fully association (全相联)  
SA = PM + FA (组关联)



① DM组成: 一个组只有行

③ SA: ~~组关联~~

一个组中有多行。

n-way SAC: 一个组有n行。

注意: 一个组内的不同行的 tag, 是并行比较判断的。

①与②比较: ③需要额外 Mux  
①的data在命中/miss前得到  
而③后

当CPU从Cache中寻找数据失败, 就会从MEM中读数据, 并替代掉

Cache中的某一行。  
(block replacement)

替换策略 { FIFO  
LRU  
LFU  
Random

(1) (tag x index) 表示了内存中的每个块

(2) index 表示映射到那个组

(3) byte selecting 表示数据位于组index中的第几个。  
起始

② FA组成:

一个组包含所有缓存行。

无需考虑组选择,

行选择与DM类似, 仍是有效位 & tag

(昂贵, 困难  $\rightarrow$  只能做小的高速缓存)

tag byte 地址 (无index)  
select

