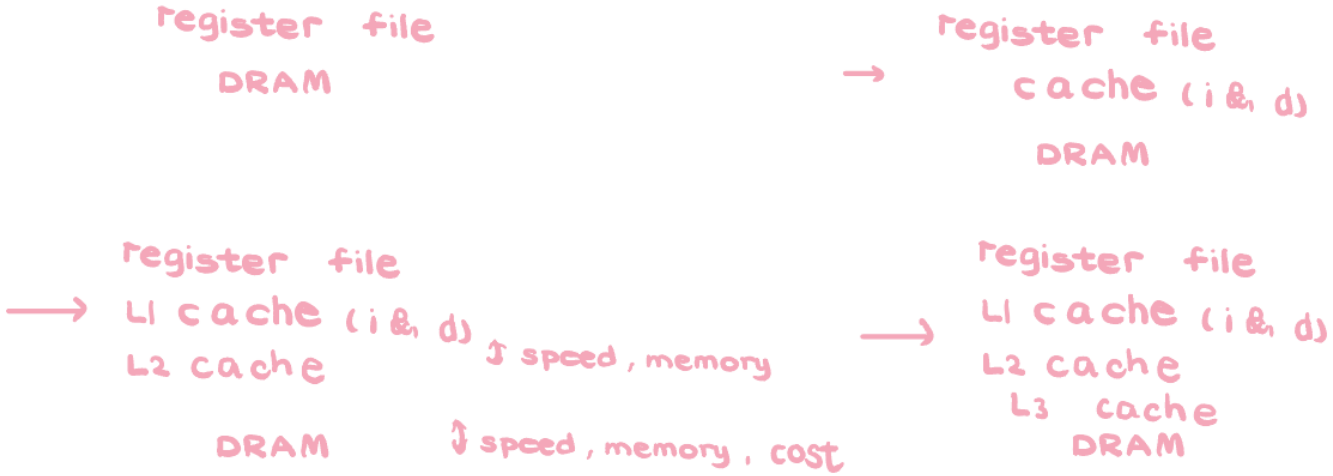
Cache Memory 回课

卢加洲

Locality

- If there’s no locality, there’s no cache

History

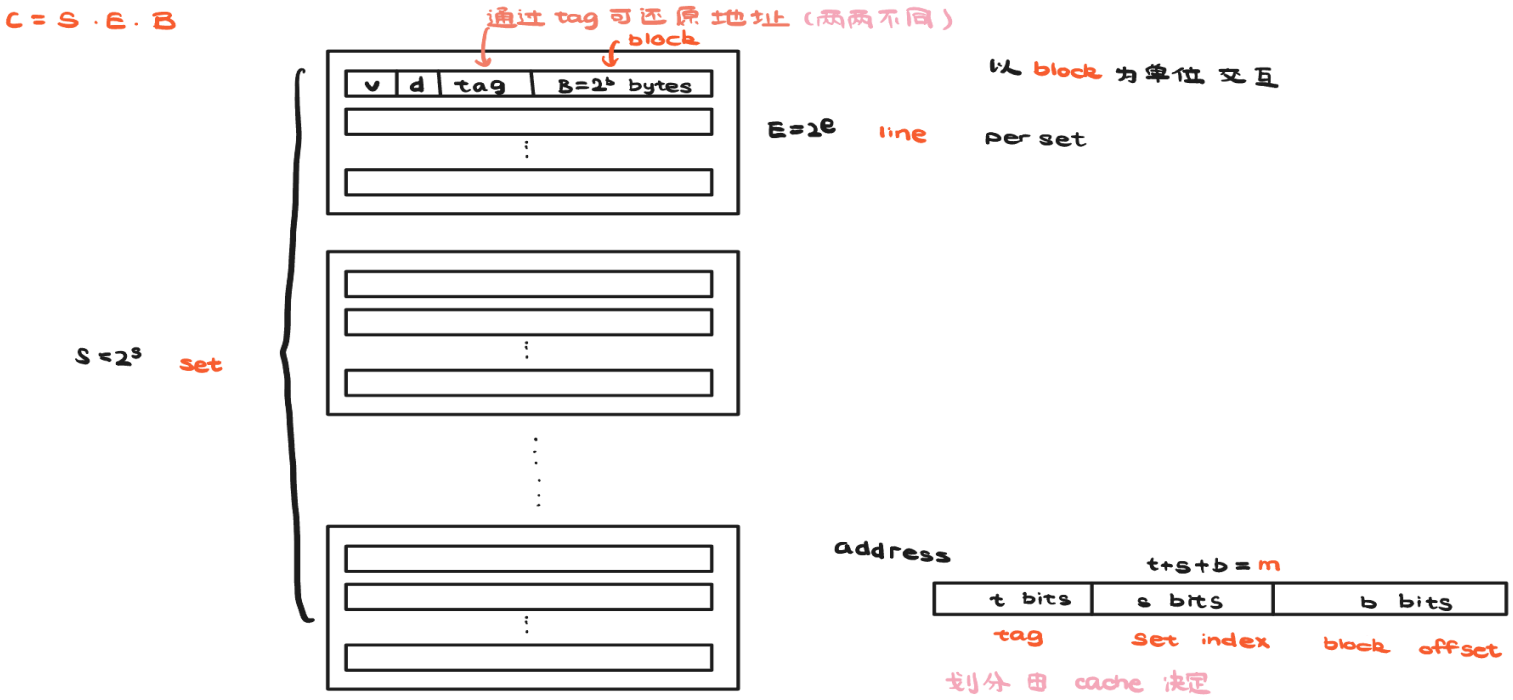

register file
DRAM
register file
cache (i & d)
DRAM
register file
L1 cache (i & d)
L2 cache
DRAM
speed, memory
speed, memory, cost
register file
L1 cache (i & d)
L2 cache
L3 cache
DRAM

Terminologies

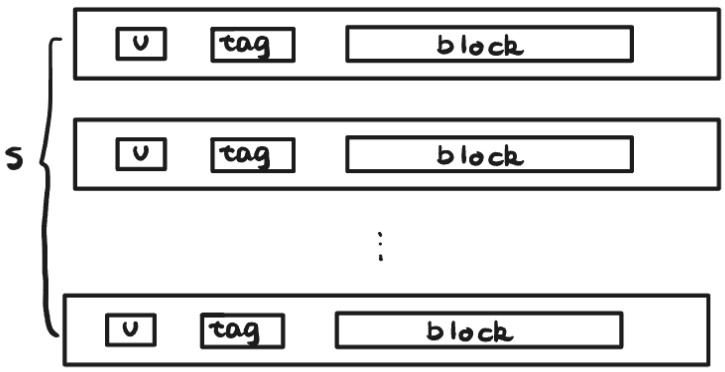
Cache 是透明的

- 由电路完成，指令不变
- 一边来说没有操作Cache的指令（例外：OS 进行Cache 清空）

Cache


C = S · E · B
通过 tag 可还原地址（两两不同）
block
v
d
tag
B=2^b bytes
以 block 为单位 交互
E=2^e line per set
S=2^s set
address
t+s+b = m
t bits
s bits
b bits
tag
set index
block offset
划分 由 cache 决定

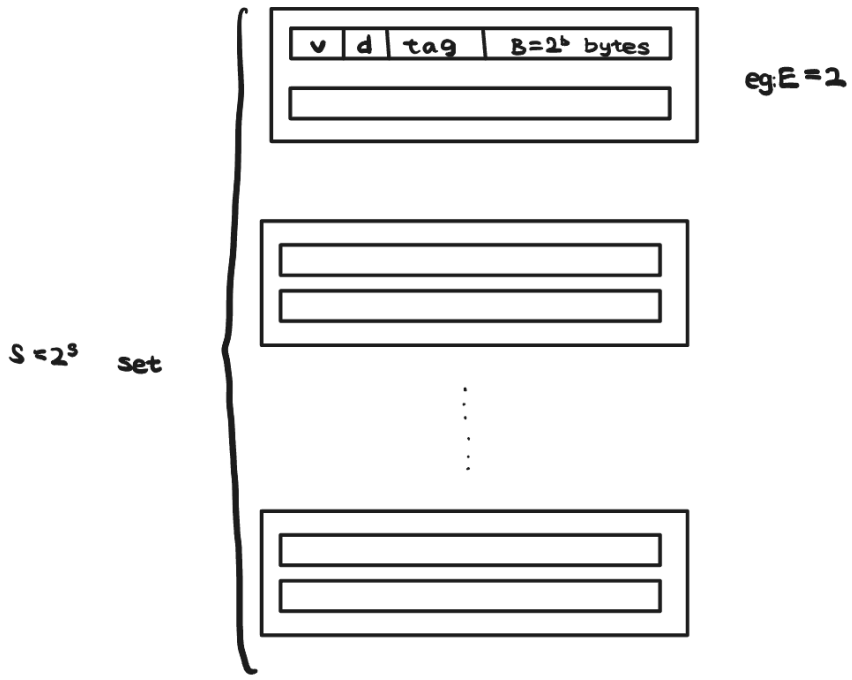
Directed Mapped Cache(E = 1)



Thrash example t = 1, s = 2, b = 1

0 [0000_2],
1 [0001_2],
7 [0111_2],
8 [1000_2],
0 [0000_2]

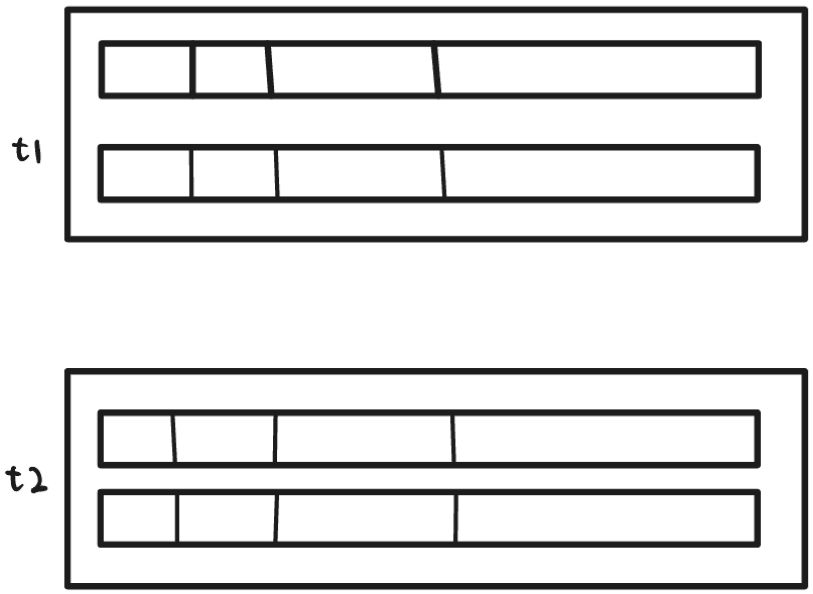
E-way Set Associative Cache



主要开销：tag 比较

Example t = 2, s = 1, b = 0

0 $[00\underline{0}0_2]$,
1 $[00\underline{0}1_2]$,
7 $[01\underline{1}1_2]$,
8 $[10\underline{0}0_2]$,
0 $[00\underline{0}0_2]$



Write policy

write through

no-write-allocate

write back :

write-allocate

add dirty bit

性能++

复杂++

Performance Judgement

Miss Rate eg: 99% 97%

Hit Time

Miss Penalty

Middle Bit Indexing

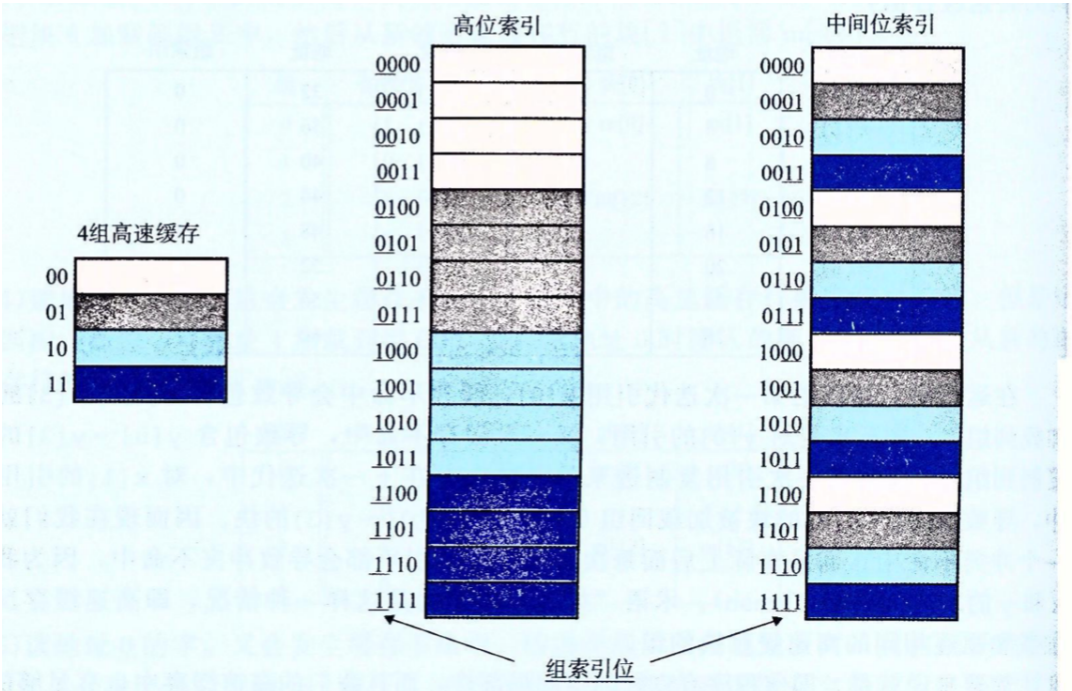


图 6-31　为什么用中间位来作为高速缓存的索引

Writing Cache Friendly Code

- Focus on inner loop
- Temporal locality
- Spatial locality

The Memory Mountain

Core i7 Haswell
2.1 GHz
32 KB L1 d-cache
256 KB L2 cache
8 MB L3 cache
64 B block size