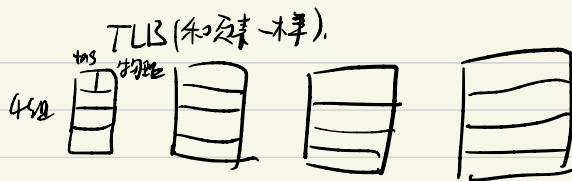
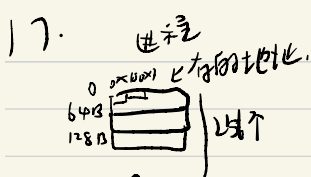


14. 2¹⁴ B.



通过页内偏移确定同一个页下不同地址

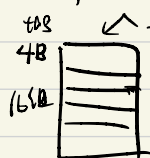
虚拟地址长14 bit, 则虚拟地址空间大小 = $2^{14} \text{ B} = 2^{14} = 256 \times 64 \text{ B}$

∴ 页表条目有256个 (只与虚拟地址相关)

页内偏移和页大小相关 4KB → 12位

64B → 6位

L1 物理缓存



← 和内存一样存储

若文+居, 如

int 数据占4B.

int 数据地址: 4B-加,

物理地址偏移

商为tag.

物理地址偏移和 cache line 相关. 字地址.

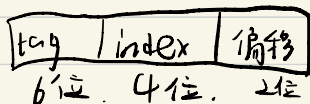
1) 索引位 10, 索引第2组, tag 为 0x05, 命中.

物理页号 0x1C, 访问实际物理地址 0x724.

2) 256个

3) 在本题中, 块偏移/位数又2位.

对于物理地址



0x724

011100 1001 00
= 1C

索引第9行, tag = 0x1C = cache line 的 tag, 第9个

所以命中缓存, 访问结果为 0xb3

18.



1)

	A	B	C	D	A	B	C	D
way0	—	A	A	C	C	A	A	C
way1	—	—	B	B	D	D	B	B
命中?	N	N	N	N	N	N	N	N

命中 0

2) 最佳替换(opt) 替换长时间内不被访问的数据来替代

	A	B	C	D	A	B	C	D
way0	—	A	A	A	A	A	B	C
way1	—	—	B	C	D	D	D	D
命中?	N	N	N	N	Y	N	N	Y

A	B	C	D	A	B	C	D	A
C	C	C	C	D	A	A	A	A
D	A	B	B	B	B	B	C	D
N	N	Y	N	N	Y	N	N	Y

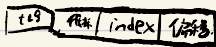
长时间, 命中率接近 $\frac{1}{3}$

19. 1) 同一缓存组若低位标签不唯一,
匹配出的缓存块不止一个, 高位还要继续再匹配。

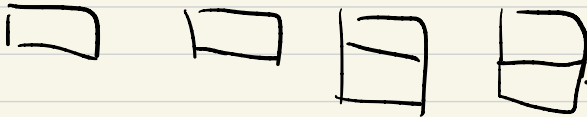
2) 减轻旧压力, 可以提前处理数据。

3) 16KB页大小, 对应页内偏移为14位。

物理地址



cache



设 index 为 x 位, 偏移为 y 位。

$$2^x (\text{组}) \cdot 4 \cdot 2^y = 2^{13} \text{ B}$$
$$\Rightarrow x + y = 11$$

低位标签至少要有 3 bit。

20-

监听一致性的优点：

实现简单：监听一致性协议较为简单，每个缓存都可以直接监视其他缓存的操作。

延迟低：由于每个缓存都能够观察到其他缓存的操作，数据的一致性可以很快地得到保证。

适用于小规模系统：在小规模系统中，监听一致性的开销相对较小，对于处理器数量较少的系统而言，可以提供较好的性能。

监听一致性的缺点：

性能开销：监听一致性需要在每个缓存中维护一份副本，随着系统规模的增大，开销会随之增加。

总线负载高：由于每个缓存都会监听总线上的操作，会导致总线负载较高，限制了系统的扩展性。

数据一致性流量增加：由于每个缓存都会广播操作，可能导致大量的数据一致性流量，增加了网络和总线的负载。

目录一致性的优点：

扩展性好：目录一致性对于大规模系统更具优势，因为它不需要在每个缓存中维护副本，减少了存储开销。

总线负载低：相比监听一致性，目录一致性减少了总线上的广播操作，降低了总线负载。

灵活性高：目录一致性可以支持更多的一致性策略，例如写回、写直达等。

目录一致性的缺点：

实现复杂：目录一致性的实现较为复杂，需要维护一个全局的目录表来记录缓存的状态和位置信息。

延迟较高：由于需要在目录表中进行查询和更新，目录一致性的延迟相对较高，对于响应时间敏感的应用可能不够理想。

缓存一致性的实现代价主要体现在以下几个方面：

硬件开销：实现缓存一致性需要在每个缓存和总线中添加额外的硬件支持，如监听逻辑、目录表等，增加了硬件成本。

性能开销：缓存一致性需要进行额外的数据一致性操作，例如缓存失效、数据更新、广播通知等，这些额外的操作会引入一定的延迟和性能开销。特别是在高并发环境下，频繁的缓存一致性操作可能会成为系统性能的瓶颈。

内存一致性：实现缓存一致性还需要考虑与主内存之间的一致性。当缓存中的数据发生变化时，需要确保这些变化及时同步到主内存，并且在其他缓存中得到更新，以保证数据的一致性。这需要一定的内存同步机制和协议，增加了实现的复杂性和开销。

调试和验证困难：缓存一致性的实现和调试比较困难。由于存在多个缓存和复杂的一致性协议，出现问题时很难确定是哪个环节导致了数据不一致的情况。需要借助调试工具和技术来进行定位和排查。