17.

1) 命中。物理地址为 0x7A4.
2) 2^14 / 64 = 8 个。
3) 0x63.

18.

1) 0%命中率。

访存地址	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？	No	No	No	No	No	No	No	No

2) FILO（栈式存储），把被使用时间较近的优先替换掉。A 一直留在 cache 内部，只要是 A 就能命中，所以 25%命中率。

19.

1) 低位标签本身就和“组号”是类似的，一般来说组和低位标签是对应的，才能比较方便地进行匹配；如果低位标签不唯一，也会导致在较前的流水线中产生较大的延时开销。
2) 引入微标签技术对通常的缓存替换策略有一定影响。常见的缓存替换策略如 LRU（最近最少使用）或 LFU（最不经常使用）等，这些策略通常基于整个缓存块的使用情况进行决策。但是，由于微标签技术仅使用低位标签进行命中判断和数据前馈，替换策略可能无法准确评估整个缓存块的使用情况。因此，在使用微标签技术的情况下，可能需要对替换策略进行适当的调整，以确保在缓存中保留最有用的数据块。
3) 页偏移 14 位，cache 指针 13 位，块偏移 2 位，低位标签至多 11 位（此时块大小为 1B，高位标签 0 位）。

20.

监听一致性是一种基于总线的缓存一致性协议。在监听一致性中，每个缓存都监视共享总线上的数据传输，并根据自己的缓存状态来决定是否读取或写入缓存。优点如下：

- 实现简单：监听一致性不需要额外的硬件支持，可以通过总线传输直接实现。
- 响应快速：由于每个缓存都能够监听总线上的数据传输，一旦检测到数据变化，缓存可以立即进行相应的更新。

然而，监听一致性也存在一些缺点：

- 总线带宽压力：由于所有缓存都需要监听总线上的数据传输，这可能导致总线带宽的限制，尤其在大规模的多处理器系统中。
- 缓存一致性消息广播：当一个处理器进行写操作时，需要通知其他缓存进行相应的更新，这可能导致大量的缓存一致性消息广播。

目录一致性（Directory Consistency）是一种基于目录的缓存一致性协议。在目录一致性中，存在一个共享的目录来跟踪每个缓存块的状态。目录记录了缓存块在哪些缓存中被缓存，以及哪些缓存具有副本。优点如下：

- 减轻总线带宽压力：目录一致性将缓存状态信息存储在共享的目录中，而不是每个缓存都监听总线上的数据传输，从而减轻了总线的带宽压力。
- 灵活的一致性策略：通过目录，可以实现更灵活的一致性策略，例如写回、写直达、无效等。

目录一致性也有一些缺点：

- 复杂的硬件支持：目录一致性需要额外的硬件支持来维护目录，并实现目录的更新和查询。
- 访问延迟增加：由于需要通过目录来确定数据所在的缓存位置，目录一致性可能会引入额外的访问延迟。

缓存一致性的实现代价体现在以下几个方面：

- 硬件开销：实现缓存一致性需要额外的硬件支持，例如总线监视、目录等，这会增加系统的硬件开销。
- 性能损失：缓存一致性协议会引入额外的访问延迟和通信开销，从而对系统性能产生一定的影响。
- 软件复杂性：为了确保正确的缓存一致性，软件需要遵循一致性协议的规则，这可能增加软件的复杂性和开发难度。