17.
(1)由于虚拟地址和物理地址的低 6 位为 offset 位，所以将 0x05a4 的第四位去除得到 00010110 也就是 0x15，TLB 中没有这个标签，因此不命中
(2)页表条目数为 2^(14–6)=256
(3)因为块数目为 16，所以 index 位数为 4 位，offset 为 2 位，由(1)可得 TLB 未命中，所以使用 0x1e4，二进制也就是 000111100100，tag 为 000111，其 2–5 位为 1001，所以查找组号为 9 的组，发现 tag 为 0x1C，二进制也就是 011100，这和物理地址的 tag 不同，所以未命中

18.
(1)

访存地址	A	B	C	D	A	B	C	D
Way0 Way1	—— ——	A ——	A B	C B	C D	A D	A B	C B
命中	N	N	N	N	N	N	N	N

(2)由于本题访问地址极有规律，以四个地址为周期，因此本缓存替换策略需要一个类似于全局历史表的功能，来记录最近访存的地址以及是否命中，假设经过很长时间，way1 和 way2 分别存储的分别是 A 和 B，此时访问 A 和 B 时显然命中，但是访问 C 时会不命中，在进行替换时，不是将 C 替换 A 和 B 中任意一个，而是根据全局历史表，将 D 替换 B，缓存中存储的是 D 和 A，这样在接下来访问 D 和 A 时均能命中，同理，接下来访问 B 时不命中，此时将 C 替换 A，缓存中存储的是 C 和 D，这样在接下来访问 C 和 D 时均能命中，同理，接下来访问 A 时不命中，此时将 B 替换 D，缓存中存储的是 B 和 C，这样在接下来访问 B 和 C 时均能命中，最后访问 D 时不命中，此时，将 A 替换 C，缓存中存储的是 A 和 B，这样在接下来访问 A 和 B 时就能命中，这样以访问 12 次为周期，1 代表命中，0 代表未命中，则规律为 110110110110，由于最多的连续数字个数为 2，因此全局历史表只要记录最近两次访存的地址和是否命中，就能找到全部的规律，这样的缓存替换策略命中率达到 66%

19.
(1)如果地位标签在同一缓存组是唯一的，那么当处理器需要访问缓存组时，其只要比较物理地址的低位标签和缓存组的低位标签是否相同，这样略去了计算 index 从而确定缓存组的过程，只需要一次地址比较就能确定对于拥有微标签的处理器低位标签比较是否命中。
(2)微标签技术会导致缓存替换策略更加关注缓存组的低位标签，由于采用微标签技术的处理器同一缓存组的低位标签相同，因此替换时会首先寻找和物理地址低位标签相同的缓存组替换。
(3)因为页的大小为 16KB，所以 offset 为 14 位，因为四路组相联缓存大小为 8KB，所以每路的大小为 2KB，也就是 index 加上 offset 的位数为 11 位，所以留给低位标签的位数为 3 位，也就是 8Byte

20.
监听一致性和目录一致性的优缺点分别为：
1. 监听一致性是指当数据库中的数据发生变化时，通过在数据库中监听器的方式及时通知应用程序，以便应用程序能够及时更新其状态。这种方式的优点是可以快速地响应数据变化，并及时更新应用程序的状态。缺点是需要占用系统资源来监听数据变化，并且可能会增加系

统的复杂度，因为需要维护监听器列表和相应的通知机制。
2. 目录一致性是指通过在数据库中维护一个中央目录来跟踪所有数据的变化，并在需要时向应用程序发送通知。这种方式的优点是可以保证数据的一致性，并且不需要额外的资源来监听数据变化。缺点是需要维护中央目录，并且需要处理大量的通知请求，这会占用系统资源。

缓存一致性的实现代价体现在：
1. 硬件成本：实现缓存一致性需要使用额外的硬件设备，如读写分离器、缓存介质等，这些硬件设备会增加系统的硬件成本。
2. 软件成本：实现缓存一致性需要编写额外的软件代码，如缓存节点之间的通信代码、缓存数据同步代码等，这些软件代码会增加系统的软件成本。
3. 性能成本：实现缓存一致性可能会降低系统的性能和吞吐量，因为需要额外的开销来处理缓存一致性问题。例如，在写时复制型缓存一致性模型中，缓存节点之间的数据同步可能会占用大量的带宽和系统资源，从而影响系统的性能和吞吐量。
4. 维护成本：实现缓存一致性需要不断的维护和调整，因为不同的缓存一致性模型有不同的实现方式和要求，需要根据不同的场景和需求进行选择和调整。