17. 假设一个使用虚拟内存和 L1 缓存的存储系统具有以下特征：

a) 内存系统按字节寻址，访存请求每次仅传递一个字节给处理器。

b) 虚拟地址长度 14 比特，物理地址长度 12 比特。

c) 页大小 64 字节，使用单级页表。

d) TLB 拥有 16 个条目，四路组相联。

e) L1 缓存物理寻址，块大小 4 字节，共 16 个组，直接映射。

现在 CPU 发起了一次对虚拟地址 0x05a4 的单字节内存加载请求，回答以下问题。

1) 若请求发起时，TLB 的部分内容如下表所示。则 TLB 是否发生命中？如果命中，此次内存访问的物理地址是多少？

组号	标签	物理页号	有效位	标签	物理页号	有效位
0	0x0B	—	0	0x1F	—	0
	0x07	0x0D	1	0x02	0x2F	1
1	0x01	0x05	1	0x05	0x0D	1
	0x14	—	0	0x2A	0x16	1
2	0x03	—	0	0x05	0x1C	1
	0x0B	0x07	1	0x00	0x1B	1
3	0x26	0x34	1	0x02	—	0
	0x19	0x2F	1	0x38	—	0

2) 该系统的页表有多少个条目？

解：(1) 由于页大小为 64B，则需用 0～5 bit 作为页内偏移

由于 TLB 有四路组相联，共 4 组，则 6～7 bit 作为 TLB 表项的索引。

0x05a4 = 0000 0101 10 0100，则索引第 2 组。0001 1 作为 Tag

则命中物理页号：0x1C，校验位为 1，故物理地址为 0001 1100 100100

= 0x724

(2) 由于 0～5 bit 是页内偏移，则 8 位的页的索引

故有 64 个页条目

3) 如果 TLB 命中，则使用 1)得到的物理地址，否则使用物理地址 0x1e4。如果 L1 缓存的内容给如下表所示，则此次访存请求是否命中缓存？如果命中，访存结果是多少？

组号	标签	有效位	块偏移			
			0x0	0x1	0x2	0x3
0	0x1F	0	—	—	—	—
1	0x05	1	0x02	0x09	0xCB	0xA3
2	0x1C	1	0x09	0x55	0x01	0x08
3	0x0D	0	—	—	—	—
4	0x1B	1	0x9B	0xEE	0xE2	0x86
5	0x2F	1	0x00	0x00	0x01	0x00
6	0x07	0	—	—	—	—
7	0x05	1	0x6F	0x23	0xAB	0xD0
8	0x16	0	—	—	—	—
9	0x1C	1	0x63	0x2F	0x1B	0x00
10	0x1C	1	0x28	0x34	0x01	0xC4
11	0x16	1	0x29	0xC8	0x56	0x99
12	0x34	0	—	—	—	—
13	0x34	0	—	—	—	—
14	0x0D	0	—	—	—	—
15	0x07	1	0xE8	0x59	0x04	0x45

解：(3) 物理地址：0x724 = 0111 0010 0100

offset = 0x0

index = 1001 — 组号为 9

tag = 01110 0

= 0x1C

则命中缓存，结果为 0x63

18. 一段程序循环往复地按顺序访问 A、B、C、D 四个地址上的数据。考虑一个拥有 2 条目的全相联缓存，回答以下问题。

1) 使用 LRU 替换策略时，填写下表。当程序长时间运行时，缓存的命中率为多少？

访存地址	A	B	C	D	A	B	C	D
way 0 way 1	— —	A —	A B	C B	C D	A D	A B	C B
命中?	N	N	N	N	N	N	N	N

2) 提出一种缓存替换策略，使得上述程序可以在该缓存中拥有最大的命中率，并计算该命中率。

(1) 长时间运行时，命中率仍为N.

(2) 替换策略：当数据进入block中，若接下来4次访问都没有使用该数据则替换。

则 A,B放入cache中后，不被替换。

长时间运行时命中率为50%

19. 一些处理器引入了“微标签”（microtag）的技术来降低组相联缓存标签匹配过程的时序压力。该技术将地址的标签部分进一步拆分为高位标签（HTag）和低位标签（LTag），在判断缓存命中与否时，控制器仅取出低位标签进行比较，将匹配的缓存块预测为一次命中并把数据前馈给处理器。在随后的剩余周期内，高位标签被取出并进一步用于判断该预测最终是否构成真正的命中。回答以下问题：

能预测

1) 低位标签在同一缓存组内通常被要求是唯一的，试说明原因。

2) 基于对 1)的讨论，简要说明该技术的引入对于通常的缓存替换策略有什么影响。

3) 考虑到虚拟页偏移和物理页偏移是一致的，为了提高访存性能，系统可以进一步要求地址的低位标签和组索引位完全位于页偏移字段内，这样低位标签的匹配过程就完全不需要经过地址翻译而可以直接进行，后续的高位标签则使用页表翻译后的结果判断是否构成真实命中。基于上述过程，对于 16KB 页大小的内存系统，一个 8KB 大小的四路组相联缓存至多可以拥有几比特的低位标签？

解：(1) 为了避免标签冲突，不同的缓存行具有相同的低位标签时，它们会被认为是同一缓存行，导致取数据、替换时混淆。

同时，发生冲突时，需要执行附加操作来解决冲突。

(2) 当没有标签冲突时，在剩余周期 高位标签会进一步判断取得的数据是否正确，

若不正确，则需要重新访问内存并替换缓存行，则第二步判断时会有替换的可能性，这会

增加替换策略的复杂性。如有近似LRU和局部LRU等。

(3) 由于页的大小为16 KB，则 offset为14位，作为标签不超过14bit

8KB的Cache有4路，每路2KB，则占11位，即Tag最多可以有3位

至多拥有3bit的组位标签

20. 监听一致性和目录一致性各有什么优缺点？简述缓存一致性的实现代价体现在哪些方面？

监听一致性：每个处理器核心都会监视（snoop）其他核心的内存访问请求操作，以确保数据一致性。

优点：①实现简单，易于设计和实现

②延迟低：每个核心可立即检查其他核心的缓存状态。

缺点：①开销大：需要不断地检查其他核心的缓存状态，增加总线和处理器核心的负载，可能降低整体性能。

②总线瓶颈：所有核心都需要监听总线上的数据传输请求，总线可能成为系统的瓶颈。

目录一致性：每个处理器在执行读写操作前，必须向目录发送请求，根据目录的响应维护一致性。

优点：①扩展性好：核心数量增加不会给总线带来过大的压力，故具有较好的扩展性。

②总线流量少：目录存储缓存数据位置信息，能减少总线上的数据传输请求。

缺点：①复杂性高：需要额外的硬件支持和目录结构来维护缓存的状态。

②延迟高：有目录发送请求和等待响应等步骤。

缓存一致性实现代价：

①多核处理器需要更大的高速缓存来存储缓存行的数据和状态信息。

②内存控制器需要额外的硬件逻辑来处理缓存操作的同步

③需要更高带宽和低延迟的总线或互连结构，以支持缓存一致性协议中的通信和数据传输。

④需要对编程模型和指令集进行适当的设计和扩展，以便开发者可以使用适当的指令或编程模式来管理缓存一致性。