**1. 连接CPU与I/O桥的总线是\_\_\_总线，连接I/O桥与主存储器的总线是\_\_\_总线，连接I/O桥与I/O 设备的总线是\_\_\_\_ 总线**

**2.**

文本

AI 生成的内容可能不正确。

**3.**

**如果直接映射高速缓存大小是4KB，并且块（block）大小为32字节，那么它每组（set）有\_\_\_\_行（line）,** **每路（way）有 行（line）。**

**4.**

文本

AI 生成的内容可能不正确。

**5.**

文本

AI 生成的内容可能不正确。

**6.**

文本, 信件

AI 生成的内容可能不正确。

**7.**

文本

AI 生成的内容可能不正确。

**8.**

某计算机系统的内存储器由cache和主存构成，cache的存取周期为40ns，主存的存取周期为260ns。已知在一段给定的时间内，CPU共访问内存4400次，其中360次访问主存。

1. Cache的命中率是多少？
2. CPU访问内存的平均时间是多少纳秒？

**9.** 假设某个计算机系统的主存容量为16MB，缓存的容量为16KB。每字块有8个字，每个字32位。设计一个四路组相联映射（即缓存每组内共有4个字块）的缓存组织，要求：

1. 画出主存地址字段中各段的位数。
2. 设缓存初态为空，CPU依次从主存第0、1、2、…、99号单元读出100个字（主存一次读出一个字），并重复此次序8次。则命中率为多少？
3. 若缓存的速度是主存速度的6倍，请问有缓存与无缓存相比，速度提高了多少倍？

**10.**

已知一个硬盘容量为1TB，由两个盘（4个盘面）构成，每个盘面由3200个磁道构成，每个磁道平均有128个扇区。由于磁记录密度的提升，新工艺维持相同的容量下，只需要一个盘（2个盘面），每个盘面的磁道数为3200，则新工艺要求每个磁道多少个扇区？

**11.**

请写出磁盘扇区访问时间的计算公式，并简单解释寻道时间、旋转时间、传送时间

**12.**

**什么是数据访问的时间局部性和空间局部性？**

**13.**

如果cache块的大小为64字节，cache总容量为1KB，请问cache的块数有多少？

**14.**

请简述直接映射、组相连映射、全相联映射的原理

**15.**

**请简单分析一下各项因素单独变化时对cache命中率的影响：cache容量大小变化、cache块大小、组的大小。**

**16.**

**如果cache块的大小为64字节，cache总容量为1KB，请问cache的块数有多少？如果采用2路组相联则32位的内存地址，将划分成标记、组索引、块内偏移三个部分各需几个bit?**

**17.**

假定某计算机的cache采用直接映射方式，和主存交换的数据块大小为1个字，按字编址，一共能存放16个字的数据。CPU开始执行某程序时，cache为空，在该程序执行过程中，CPU依次访问以下地址序列：2、3、11、16、21、13、64、48、19、11、3、22、4、27、6和11。

（1）每次访问在cache中命中还是缺失？试计算访问上述地址序列的cache命中率。1/16

（2）若cache容量还是16个字，而数据块大小改为4个字，则上述地址序列的命中情况又如何？说明块大小和命中率的关系。1/4

**18.** 假定某计算机的数据cache采用全相连映射方式，即内存的任意一个块可以被放置到cache中的任意位置。主存与cache交换的块大小为2个字（2B），数据cache一共可以存放4个字。当cache发生冲突时，选取最近最久不使用（LRU）的那个块被换出。数组按行优先方式存储，请分别计算代码段1和代码段2的Cache命中率，讨论哪一个代码对cache更友好，并分析cache不友好代码的原因。

文本, 信件

AI 生成的内容可能不正确。 文本, 信件

AI 生成的内容可能不正确。

代码1 代码2

**19.**

表格

AI 生成的内容可能不正确。

**20.**

文本

AI 生成的内容可能不正确。

**21.**

请简单比较直接映射和全相联映射在硬件实现上的复杂程度的差异

答案：

**1. 系统，内存，I/O**

**2. D**

**3. 1 128 : 容量=路容量\*路数=cache 行大小\*Set 数\*路数->set 数 =4KB/32B=2^7=128**

**4.** **B**

**5.** 答案：C 数组本身只能体现出空间局部性

**6.** **D**

**7.** **A**

**8.** **参考答案：**

**h = (4400-360) / 4400 = 91.8%**

**Ta = h\*Tc + (1-h)\*Tm = 57.6s**

**9.**

文本, 信件

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。

**10.** **256**

**11.**

**访问时间=寻道时间+延迟时间+传输时间**

**寻道时间，也称寻找时间：磁头移动到指定磁道需要的时间**

**延迟时间：磁头定位到某一磁道的扇区所需要的时间**

**传输时间：从磁盘读出或者写入经历的时间**

**12.**

**时间局部性：最近被访问的数据或指令在未来可能还会被访问。**

**空间局部性：最近访问地址附近的区域在不久可能还会被访问**

**13.** cache的块数

**14.** 直接相联(Direct)：每个主存块映射到Cache的固定行。

直接映射是一种简单的映射方法，将主存地址空间分成多个块，并将缓存分为相同数量的行。每个主存块只能映射到缓存中的特定行。具体的映射规则是，主存地址的一部分用于标识缓存中的行号，另一部分用于标识行内的偏移量。当需要访问某个主存地址时，先计算出对应的行号，然后检查该行是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉缓存中的某一行。

组相联(Set Associate)：每个主存块映射到Cache固定组中任一行。

组相连映射是在直接映射的基础上进行改进的一种映射方法。相比直接映射，组相连映射在缓存中引入了多个行的集合，称为组。每个主存块可以映射到多个组中的某一行。具体的映射规则是，主存地址的一部分用于标识组号，另一部分用于标识行内的偏移量。当需要访问某个主存地址时，先计算出对应的组号，然后在该组中查找是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉该组中的某一行。

全相联(Full Associate)：每个主存块映射到Cache的任一行。

全相联映射是最灵活的映射方法之一。在全相联映射中，主存地址可以映射到任意的缓存行。具体的映射规则是，主存地址的所有位都用于标识缓存中的行号。当需要访问某个主存地址时，需要遍历整个缓存以查找是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉缓存中的某一行。

**15.**

Cache容量大小：容量提高会提升命中率，增加命中时间。

Cache块大小：块大小提高有利于提高命中率，但随着块数量减少，命中率也会下降。

组大小：较大的组降低了高速缓存由于冲突不命中出现抖动的可能性。

**16.** 内存地址32位，块内6 bits，cache有24= 16块，2路组相联，则cache有8组（3bits），每组内有2块。

23 3 6

|  |  |  |
| --- | --- | --- |
| 标记tag | 组索引 | 块内偏移 |

文本, 信件

AI 生成的内容可能不正确。

**17.** **1/16，1/4**

**18.** 第二个代码对cache更友好。 第一个代码的数据访问没有好的局部性

代码1 的命中率为0，代码2 的命中率为50%

**19.** **平均旋转时间=**0.5\*1/7200\*60\*1000ms

**数据传输时间=**1/7200\*60\*1000/600ms

**一个扇区的平均访问时间=平均寻道时间+平均旋转时间+数据传输时间**

**20.**

文本

AI 生成的内容可能不正确。

**21.** 全相联映射需要进行更多的标记位比较，硬件实现更复杂