1.请阅读以下代码

int main(){

int x = 0；

float y = 0.0;

while((y - x < 1) && (y - x > -1)){

x += 1;

y += 1;

}

printf("%d %f\n", x, y);

}

问最终的输出为：

（提示1：我们这里认为int型变量与float型变量运算会把int型转化为float型再进行运算）

（提示2：2^24 = 16777216）

（提示3：本题中的舍入方式为向偶数舍入）

A.16777218 16777217.000000

B.16777218 16777216.000000

C.16777217 16777216.000000

D.16777217 16777215.000000

2.现定义了如下变量：

int x, y, z;

float fx, fy, fz;

则以下表达式不恒为真的是（其中fx,fy均不为无穷或非数）：

A.x \* y == y \* x

B.(x + y) \* z == x \* z + y \* z

C.fx + fy == fy + fx

D.(fx + fy) \* fz == fx \* fz + fy \* fz

3.下列选项中是合法x86-64汇编指令的是（）

A.movq %rax, $1

B. xorq %rsp, %rsp

C. movq (%rax), (%rbx)

D. movq (%rax, %rbx, 3), %rcx

4.现假定所有函数均使用%rbp寄存器描述栈帧，并且函数体进入时一定以pushq %rbp; movq %rsp, %rbp开始，函数体退出前一定以leaveq; retq结束。有编译器内置函数 \_\_builtin\_return\_address(1)表示获取调用当前函数的函数的返回地址，举例来说，若有如下代 码：

long foo(void)

{

return (long) \_\_builtin\_return\_address(1);

}

同时函数bar()调用了函数foo()，则函数foo()的返回值是函数bar()的返回地址(即bar()调用 完成后控制流转向的地址)，保证bar()和foo()都不是内联函数，则将函数foo()编译，生成的 汇编指令可能是（）

A.

foo:

pushq %rbp

movq %rsp, %rbp

movq 8(%rbp), %rax

leaveq

retq

B.

foo:

pushq %rbp

movq %rsp, %rbp

movq 0(%rbp), %rax

movq 0(%rax), %rax

leaveq

retq

C.

foo:

pushq %rbp

movq %rsp, %rbp

movq 0(%rbp), %rax

movq -8(%rax), %rax

leaveq

retq

D.

foo:

pushq %rbp

movq %rsp, %rbp

movq 0(%rbp), %rax

movq 8(%rax), %rax

leaveq

retq

5.下列关于RISC和CISC的描述中，正确的是：

A.在RISC指令集的发展过程中，其指令数量始终少于100条。

B.CISC指令集中的指令比RISC指令集更复杂，因此其指令长度总是比RISC指令集中的指令要长。

C.早期的RISC指令集没有条件码，对条件检测来说，要用明确的测试指令，这些指令会将测试结果放在一个普通的寄存器中。

D.RISC指令集中的所有过程都需要进行内存引用。

6.在Y86-64 PIPE处理器中（不考虑数据前递），以下哪个指令序列会造成数据冒险？

A.

irmovq $10, %rdx

addq %rdx, %rax

B.

irmovq $10, %rdx

nop

addq %rdx, %rax

C.

irmovq $10, %rdx

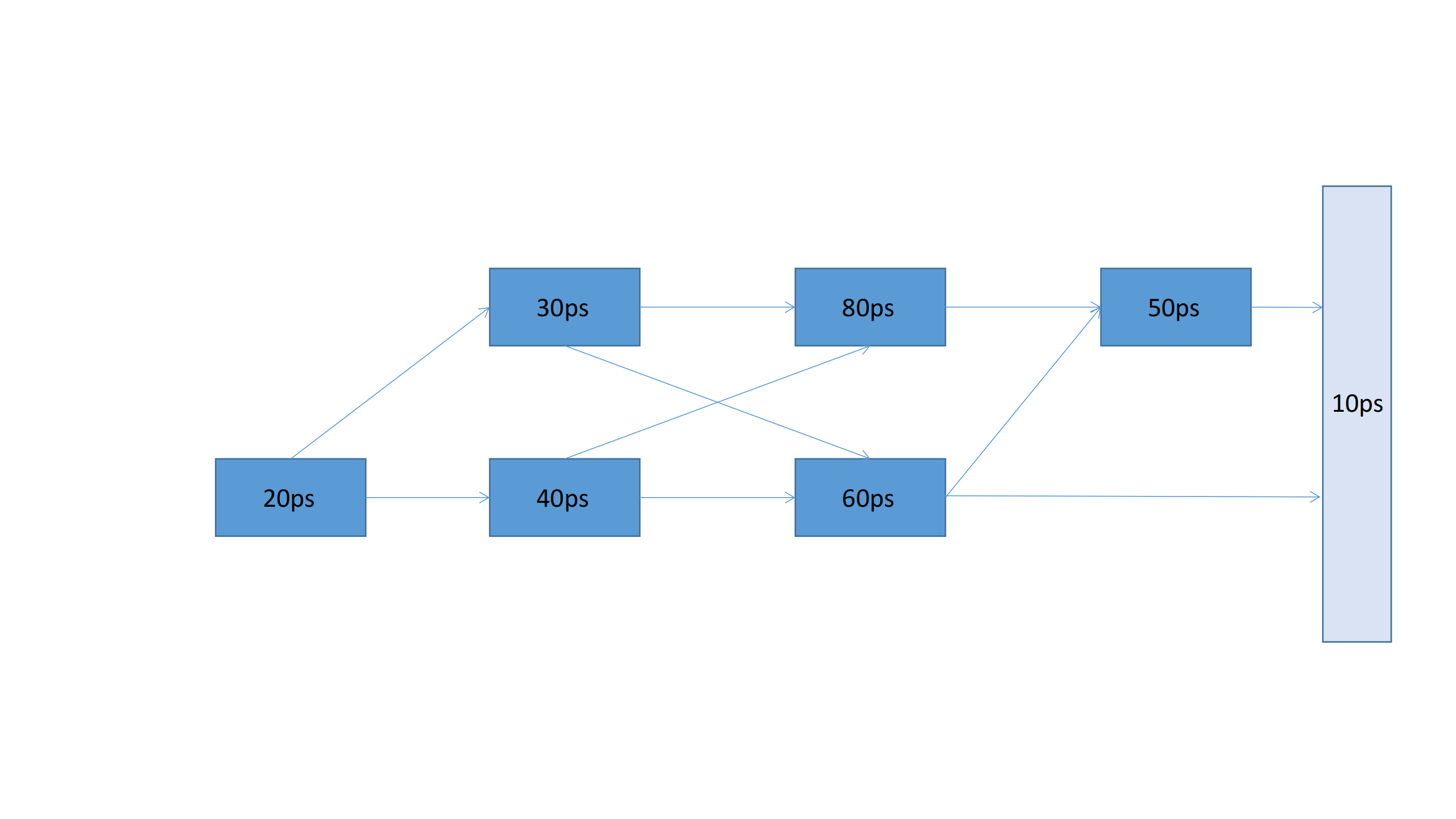
nop

nop

addq %rdx, %rax

D. 以上三个选项都会引发数据冒险

7.根据下图所示的流水线结构，判断错误的选项是？

A.当前流水线的延迟为(20ps + 40ps + 80ps + 50ps + 10ps)=200ps，吞吐率为5GIPS。

B.无法通过只插入一个寄存器的方式来使这个流水线变为二级流水线。

C.倘若只能插入两个寄存器(延迟均为10ps)，那么该流水线处理每条指令的平均时间最多减少20ps (假设每条指令平均使用一个周期的时间)。

D.流水线的级数越深，处理器就一定能获得越好的性能。

8.以下关于程序设计优化的说法，错误的是：

A.循环展开有助于进一步变换代码，减少关键路径上的操作数量。

B.分离多个累积变量计算可以提高并行性。

C.大多数编译器会改变浮点数的合并运算（如加法和乘法）顺序以提高程序性能。

D.循环展开的程度增加并不一定能改善程序运行效率，反而会变差。

9.以下关于存储设备与存储器层次结构的说法，正确的是：

A.一个有5个盘片，每个面10000条磁道，每条磁道平均有400个扇区，每个扇区有512个 字节的硬盘容量为20.48GB。

B.SRAM的存取速度快，但是断电后数据会丢失，DRAM的存取速度较慢，但断电后数据不 会丢失。

C.在存储器层次结构中，上一层存储的信息一定是下一层存储的信息的一个子集。

D.基于缓存的存储器层次结构只利用了程序的时间局部性，没有利用程序的空间局部性，因为 缓存不会存储我们没访问过的元素。

10. 布局是数字芯片设计自动化流程的必要步骤，下面哪一项不是布局算法需要优化的目标？

A. 互连线长（Wirelength）

B. 可布线性（Routability）

C. 时序（Timing）

D. 逻辑深度（Logic depth）

答案与题解：

1.答案：B。由代码可以知道，循环会在x与y的值首次不相等的时候终止，这当然会在y首次发生舍入附近。由于当y达到值16777216后就不会发生变化，因此之后就只考察x的变化。x在值为16777217时类型转换为float会向偶数舍入至16777216，因此最终x的值为16777218。

2.答案：D。大数吃小数。

3.答案：B。立即数不能作为mov指令的目标，A错误；mov指令无法操作两次内存，C错误；内存寻址比例因子必须是1、2、4、8，D错误。

4.答案：D。在foo()中读取0(%rbp)可以获得foo()开头保存的%rbp，也就是外层函数bar()的栈帧地址，向上偏移8即可获得函数bar()的返回地址。

5.答案：C

解析：A选项，RISC指令集有复杂化的趋势，早期RISC通常指令数量小于100条；B，CISC的指令是不定长的，有些长度比RISC短；C正确；D，CISC一定会将返回地址压入栈中，不可能避免内存引用。

本题考查对两种指令集的理解。

6.答案：D。要想不引起数据冒险，需要改变数据的指令在使用数据的指令进入译码阶段时已经完成写回阶段。因此至少需要插入3个nop指令。

书本P295

7.答案为D。送分。

8.答案：C。浮点数运算没有结合律，大多数编译器不会尝试优化。

9.答案：A。B中DRAM掉电后也会丢失，C中上层存储的内容可以独立于下层，例如加载到寄存器中的值不一定在内存中存在。D缓存利用了空间局部性，会把访问元素在同一块内的元素全部取到缓存里。

10．答案为D，见讲座PPT25页，凭直觉也可以获得答案：布局算法会影响互连线长和走线方式，因此A和B需要考虑，线长会影响延时，所以C 时序也需要考虑。D的逻辑深度在布局时不变。因此正确答案选D

**第二题** 请结合教材第六章”存储器层次结构”的有关知识回答问题**(10分)**.

1. **(2分)**高速缓存(cache)的先进先出替换策略(FIFO)指的是在发生缓存不命中时,最先进入高速缓存的行(cache line)将最先被替换出.考虑两种遵循先进先出替换策略且块大小(block size)相同的全相联高速缓存C1和C2.其中C1是3路全相联的,C2是4路全相联的.假设初始情况下C1,C2所有行的有效位都为0.则让它们分别连续访问标记(tag)为0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4的行之后, C1发生的缓存不命中次数**(1)**\_\_\_\_\_\_\_\_C2发生的缓存不命中次数**(填“>”“=”或“<”)**.
2. **((2)(3)(4)每空2分,(5)(6)每空1分)**已知某单核处理器有L1,L2,L3三级高速缓存且遵循先进先出替换策略,你希望通过触发行驱逐(cache line eviction)的方法测量该处理器的L1数据缓存(L1 d-cache)的相联度,实验前已知L1 d-cache的容量在16KiB到64KiB之间(含16KiB和64KiB)且字节数为2的幂,且已知高速缓存块的字节数小于1KiB.你的实验步骤如下：

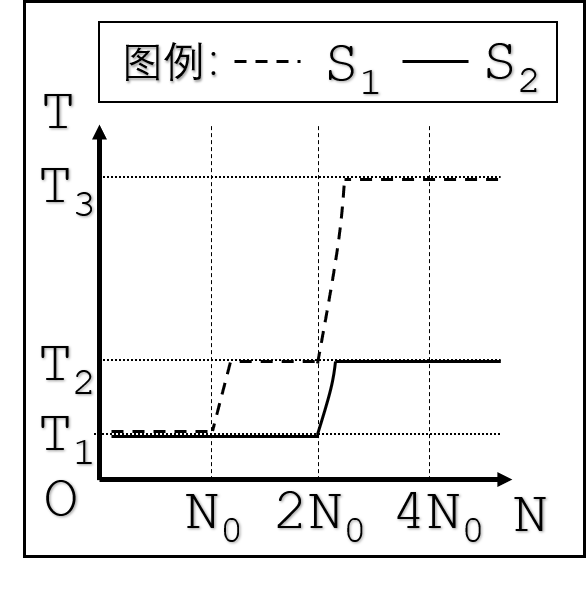
**Step #1:** 开辟一块足够大(如8MiB)的内存空间.

**Step #2:** 逐步调整访问的步长S(S依次取1KiB, 2KiB, 4KiB, …, 32KiB,64KiB)和访问的元素数量N(N依次取1, 2, 3, …, 127, 128).对于给定的S和N,连续两次从所开辟内存的起始位置开始,以S为步长顺序访问N个内存地址.记录第二次访问时平均一个内存地址的访问时间T.

**Step #3:** 多次实验取平均值以减小误差.作图并分析实验结果.

这里是给定S和N后访问内存的一个例子：当S=64KiB,N=6时,先按顺序访问一遍所开辟内存空间的第0B, 64KiB, 128KiB, 192KiB, 256KiB, 320KiB处的一个字节的值.清空寄存器,接着计时并第二遍按顺序访问所开辟内存空间的第0B, 64KiB, 128KiB, 192KiB, 256KiB, 320KiB处的一个字节的值.访问完成后立即停止计时,并记录T = (结束计时的时刻 - 开始计时的时刻) ÷ 6.0.

1. **Step #2**中,给定S和N后第一遍访问N个内存地址的过程被称为暖身(warm up),这是为了避免教材中所提到的三种缓存不命中之一的**(2)\_\_\_\_\_\_\_\_**不命中带来的影响.

****

1. 假设访问内存的过程都在高速缓存中顺次进行,且不考虑预取(prefetch)机制的影响.你选取了步长为S1, S2时的部分实验结果如图所示(此图为示意图).你注意到访存速度越快,则访问单个元素的时间越短.因此从图中可以看出T3最接近**(3)\_\_\_\_\_\_\_\_(填“L1”“L2”或“L3”)**级缓存的访存时间,S1**(4)\_\_\_\_\_\_\_\_**S2**(填“>”“=”或“<”).**
2. 在**(b)**的条件下, 进一步推理可知该处理器的L1 d-cache的容量(capacity)为**(5)**\_\_\_\_\_\_\_\_**(用代数式表示,S1, S2, N0为已知量,下同.注意计算容量时不考虑有效位和标记位)**,相联度为**(6)**\_\_\_\_\_\_\_**.**

**标准答案(每空2分)：**

**(1) ＜**

**(2) 冷/强制性**

**(3) L3**

**(4) ＞**

**(5) 2·N0·S2**

**(6) N0**

**评分标准:**

1. **写“小于”的酌情扣1分,回答其他答案（如≤）的不得分.**
2. **写“冷”“强制性”也正确.写英文也正确.字写错或有拼写错误的,若不影响意思表示不额外扣分.**
3. **写“L3”也正确,写“三级”的酌情扣一分.**
4. **写“大于”的酌情扣1分.**
5. **角标书写不规范,如写成“2·N0·S2”,若不影响意思表示不额外扣分.**
6. **角标书写不规范,如写成“N0”,若不影响意思表示不额外扣分.**

**解析:本题设简单题8分,难题2分.预测平均分7.5分.**

1. **考察高速缓存替换模拟,属简单题,预测正确率90%.应当注意FIFO和LRU替换机制的区别.**
2. **考察缓存不命中的类型,属简单题,预测正确率90%.**
3. **考察读图分析数据的能力,属简单题,预测正确率90%.因为L3高速缓存访存速度最慢,因此平均访问时间最长.**
4. **考察读图分析数据的能力,属简单题,预测正确率80%.由于N=2N0时S1发生了L2到L3高速缓存的访存切换,S2发生了L1到L2高速缓存的访存切换,故有S1＞S2.**
5. **考察对多路组相联高速缓存替换原理的理解,属难题,预测正确率25%.**

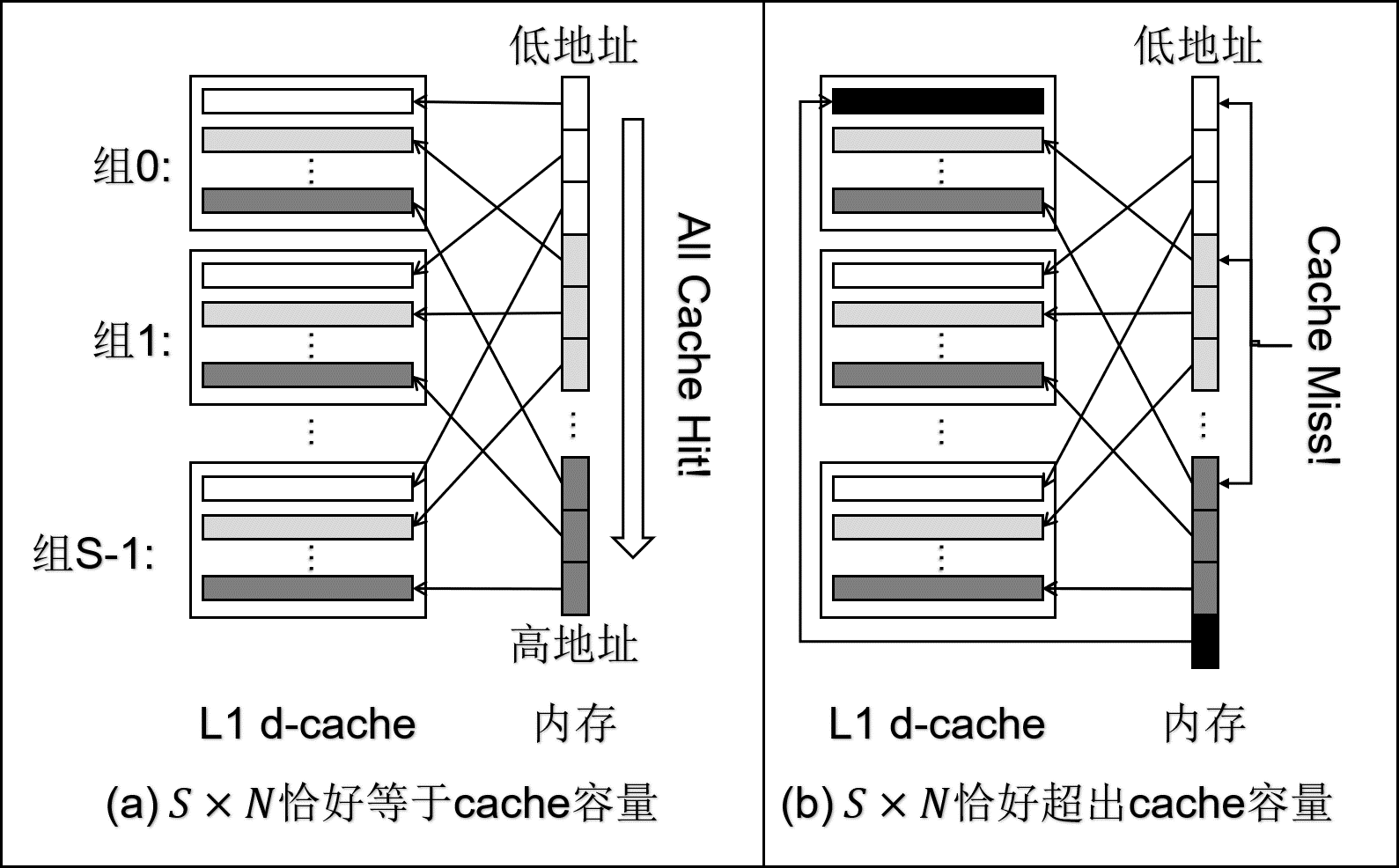


图2-2

**(i) 虽然步长S是从1KiB开始测量的,但我们不妨先考虑步长更小,即S恰好与L1 d-cache缓存块大小相等的情形,如图2-2(a)所示.不失一般性,我们假设所开辟内存空间的起始地址对应到L1 d-cache的组0.当恰好为L1 d-cache的容量时,L1 d-cache恰好缓存了所有需要访问的值.此时,所有的访存都是在L1 d-cache中进行的.现在保持S不变,把N调整为N+1.如图2-2(b)所示.此时L1 d-cache不得不在组0中替换掉一行.根据先进先出的替换原则,被换出的行恰好是第一个被加载进L1 d-cache的行.所以,在第二遍访问所开辟内存空间的起始地址时,会遇到一次缓存不命中.而加载该地址的过程又会替换掉第一遍访问时第二个被加载进组0的行,使第二遍访问它时也是不命中的.以此类推,你发现访问所有对应到组0的地址都是不命中的.频繁的L2 cache访存导致平均访存时间的剧烈上升.  
(ii) 再考虑保持不变,步长从S变成2S, 4S……时的情形.如图2-3所示,你发现当步长翻倍时,虽然访存次数减少到原来的一半,但是L1 d-cache缓存的有效的值的数量可能也会缩小到原来的一半.这使得平均访存时间可能是不变的.**

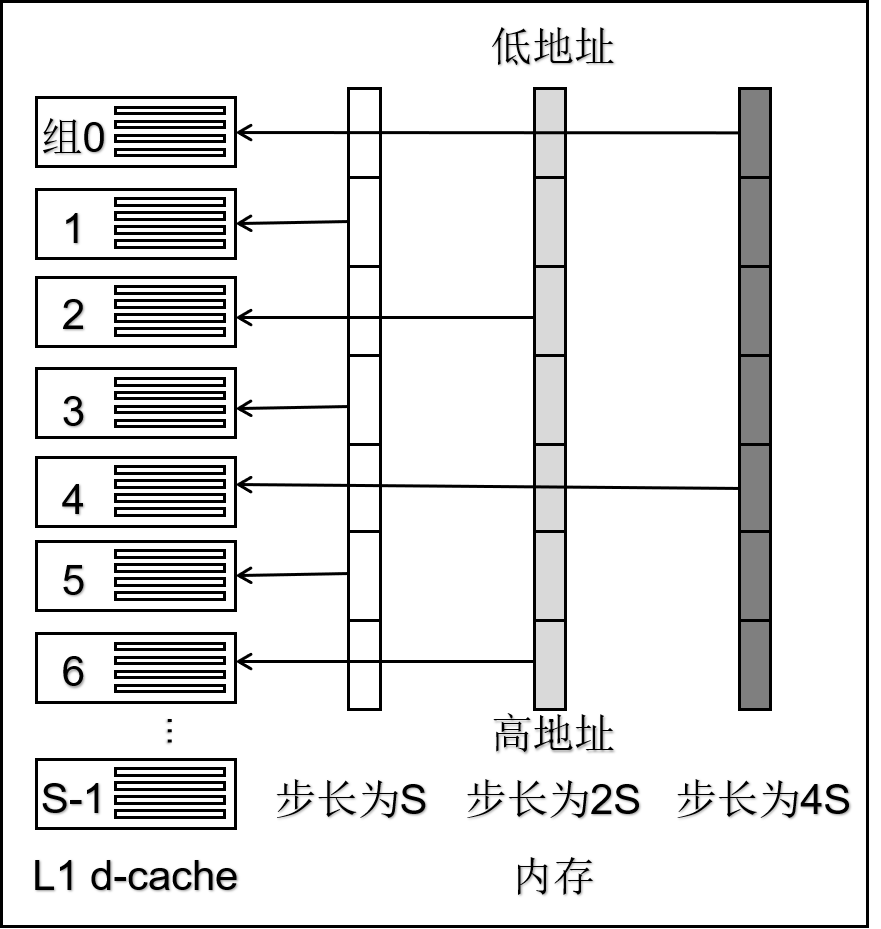


图2-3

**假想S从B逐步增长到了S2,而N=2N0处平均访存时间发生剧烈的上升,这说明L1 d-cache的容量为2·N0·S2.**

1. **考察多路组相联高速缓存相联度的性质,属难题,预测正确率25%.如果高速缓存是直接映射的,那么步长每增加一倍,则访存时间T的转折点横坐标应该变为原来的1/2,但如果高速缓存的相联度不为1,那么转折点横坐标在到达相联度的时候就不会再减小.从图中可以看出,步长S1至少是S2的4倍（因为S=S2,N=4N0时,平均访存时间T没有发生剧烈变化）.但S1的从T1到T2的转折点横坐标只是S2转折点横坐标的1/2.这说明L1 d-cache的相联度为N0.**

**命题人：李浩雨,陈奕奇**

**2021年12月4日**