6. 试简要分析为什么缓存一般使用地址的中间位作为组索引、高位作为标签，而不是高位作为组索引，中间位作为标签？

地址低位变化比高位快，高位相对低位稳定被，将中间位编码成组索引位，将相邻内存块映射到不同缓存组，同时还能共用相同标记位。

7. 一些真实的缓存系统在设计时会有意让地址的组索引和块内偏移的总位数与虚拟内存地址系统系统的页偏移位数相同，试分析这样做有什么好处。

①简化地址转换：使用相同的位数来进行地址转换操作，可以简化地址转换的逻辑和实现。

②提高地址映射效率：由于组索引和～相同，地址的不同部分可以在同一步骤处理，提高地址映射效率，减少处理时间。

③减少冲突：如果组索引和块内偏移的总位数不同，可能会导致不同的虚拟地址映射到同一缓存组中，增加了缓存冲突的可能性。

④提高缓存利用率：相同组索引可以映射到不同的块内偏移，存储更多不同的数据块，提高缓存利用率。

8. 缓存作为加速处理器访存的手段，依赖于处理器访存的局部性原理。假设一个由L1缓存和主存构成的存储系统。不同情况下的访问延时为：缓存命中时1周期、缓存缺失时110周期、禁用缓存总是直接访问主存时105周期，则：

1) 如果使用L1缓存，且程序的平均缓存缺失率为3%。则存储系统平均访问延时为多少？

$T = 1 + 3\% \times 110 = 4.3$ cycle

2) 如果一个具有完全随机数据访问性质的程序，该程序持续地访问一个1GB数组中的随机位置。假设主存足以放下完整数组，L1大小为64KB，则存储系统平均访问延时为多少？

1GB=1024KB 故缺失率为 $\frac{1024-64}{1024} = 93.75\%$

$T = 1 + 93.75\% \times 110 \approx 104$ cycle < 105 cycle

3) 利用a和b结果，说明局部性原理如何影响处理器的访存性能。

程序连续访问数据或指令往往在物理存储器中也是连续存放的，故利用缓存预存放可以减少访存延迟，但如果访问数据远大于缓存大小，则局部性原理近乎失效。

4) 程序的平均缓存命中率需要满足什么条件，才能让本例中的存储系统在使用L1时获得性能收益？

$T = 1 + \eta \times 110 < 150105$ $\eta < \frac{104}{110}$ 则 $1 - \eta > \frac{6}{110} \approx 5.5\%$ 则命中率大于5.5%即可

9. 根据给出的不同缓存配置，补全下表中缺失的字段。

编号	地址位数Bit	缓存大小KB	块大小Byte	相联度
1	32	4 (2^2)	64 (2^6)	2
2	32	4	64	8
3	32	4	64	全相联
4	32	16 (2^{14})	64	1
5	32	16	128 (2^7)	2
6	32	64 (2^{16})	64	4
7	32	64	64	16
8	32	64	128	16

组数量	组索引位数Bit	标签位数Bit	偏移位数Bit
32	5	21	6
8	3	23	6
1	0	26	6
2^8	8	18	6
2^7	7	18	7
2^{10}	10	16	6
2^{10}	10	16	6
2^9	9	16	7

10. 虽然较大的缓存通常有较低的缺失率，但更大的缓存一般也带来更高的命中延时，最终的性能受什么因素的影响决定。假设两个具有L1缓存的存储系统：系统A使用8kB直接映射缓存，命中延时为0.22ns，缓存缺失率为P_1；系统B使用64kB四路组相联缓存，命中延时为0.52ns，缓存缺失率为P_2则：

1) 如果缓存的缺失代价均为额外100ns，则P_1和P_2满足什么条件时，系统A的平均内存访问时间优于系统B？

$T_A = 0.22ns + P_1 \times 100ns$；$T_B = 0.52ns + P_2 \times 100ns$

$T_A < T_B$ 则 $P_1 - P_2 < 3 \times 10^{-3}$ (0.3%)

2). 如果缓存的缺失代价分别是各系统命中延时的K倍，则P_1和P_2满足什么条件时，系统A的平均内存访问时间优于系统B？

$T_A = 0.22ns + P_1 \times k \times 0.22ns$；$T_B = 0.52ns + P_2 \times k \times 0.52ns$

$T_A < T_B$ $\Rightarrow (0.22P_1 - 0.52P_2) < \frac{0.3}{k}$ 此时A优于B.

11. 根据数据缓存共有16个块，块大小为64字节。若有一系列发往数据缓存的请求，它们的块地址(块地址是已经去除块内偏移后的内存地址)为：0x1001、0x1005、0x1021、0x1045、0x1305、0x2ee5、0xff05。假设数据缓存初始为空，讨论当此数据缓存为直接映射、2路组相联、4路组相联和8路组相联的情况下，缓存发生块替换的次数。

块内偏移：6位

直接映射：索引：4位

0x1001 mod 16 = 1　　0x1005 mod 16 = 5　　0x1021 mod 16 = 1

0x1045 mod 16 = 5　　0x1305 mod 16 = 5　　0x2ee5 mod 16 = 5

0xff05 mod 16 = 5

替换5次

2路：3次　4路：1次　8路：0次

12. 增加缓存的相联度通常可以降低缺失率，从而提升性能。但对于特殊的程序，情况可能相反。假设两个块大小均为16字节、总容量均为256字节的缓存：缓存A为2路组相联，缓存B为直接映射。两个缓存均使用LRU替换策略。分别计算运行下述程序时，使用上述两种缓存的缺失率。(初始时缓存为空，i和j均储存于寄存器中，数组array在内存中的起始地址为0)。

```
#define N 100
int32_t array[96];
for(int32_t i=0; i<N; i++){
    for(int32_t j=0; j<96; j++){
        array[j] = i*j;
    }
}
```

分析：对96个数组单元的100次写(读)　　块：$\frac{16}{4}=4$每个数据　$\frac{96}{4}=24$

256÷16=16　有16个块块大小16B(直接映射).　　16+8=24 直接命中　$n=(3\times 8\times 100+8\times 99\times 8\times 4)$.

96÷16=5　　$R=n\div N=0$　　$N=96\times 100$

两路组相联 有8个组 8×2 R=0 故两种下缺失率均为1。　$R=\frac{n}{N}=83.25\%$

两路组相联：N不变 $n=3\times 8\times 100\times 3$　$R=75\%$

0~7　0~7　0'~7'　0~7

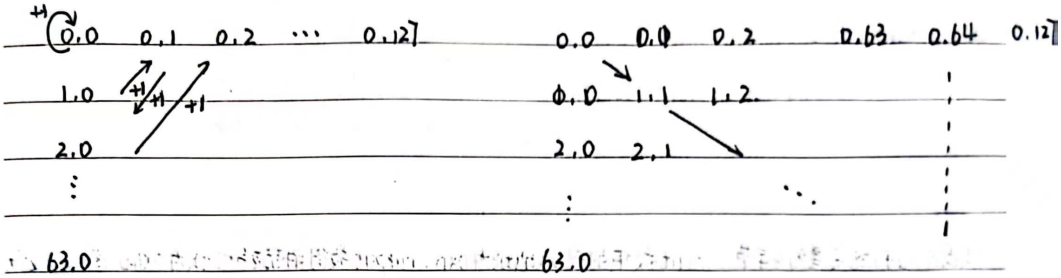
0'~7'　0~7　0'~7'

13. 考虑如下所示的C语言代码片段：

```
for(int i=0 ; i<64 ; ++i){
    for(int j=0 ; j<128 ; ++j){
        A[j][i] = A[j][i]+1;
    }
}
```

}在内存中数据的组织方式为A[p][q]与A[p][q+1]相邻。请从利用缓存空间局部性的角度，优化以上代码，不可更改代码实现的功能。

分析：A[0][0] = A[0][0]+1 ; A[0][1] = A[1][0]+1; … A[0][127] = A[127][0]+1



优化后：

```
for(int i=0 ; i<64 ; ++i){
    for(int j=i ; j<128 ; ++j){
        A[i][j] = A[j][i]+1;
        if(i!=j && j<64)
        A[j][i] = A[i][j]+1
    }
}
```

```
for(int j=0 ; j<128 ; ++j){
    for(int i=0 ; i<64 ; ++i){
        A[j][i] = A[j][i]+1;
    }
}
```

14. 仍然考虑题13中的代码，假设系统仅在对数组A发生读写时访问缓存，其他变量均全部位于寄存器中。缓存初始为空。回答以下问题：

1) 假设存在一个4KB大小的直接映射缓存，块大小32字节。则优化前和优化后代码的缓存缺失次数分别为多少？

2) 假设存在一个4KB大小的全相联缓，块大小32字节，使用FIFO替换策略。则优化前和优化后代码的缓存缺失次数分别为多少？

3) 对于块大小32字节的直接映射缓存，如果希望系统除了必要的强制缺失外不出现任何其他缺失，则优化前和优化后代码分别至少需要多大的缓存容量？

R L
128×64

1) 2^{12}B 32B 则有2^7组（128组）. N=2×128×64 n=128×64 miss=1/2 次数 64×16=1024

优化后 128=2×64 miss=1/2 次数 1024次

2) 次数前后为128×64 1024次 64次

3) 64×128

15. 考虑如下实现n阶对称矩阵转置功能的C语言代码片段：

```
int input[4][4], output[4][4];
for(int i=0;i<4;++i){
    for(int j=0;j<4;++j){
        output[j][i]=input[i][j];
    }
}
```

00 01 02 03
10 11 12 13
20 21 22 23
30 31 32 33

假设系统中int变量为4字节，input数组的起始地址为0x0，output数组的起始地址为0x40.假设上述程序运行在一个有在L1缓存的系统中，该缓存总大小32字节，块大小16字节，直接映射，缓存策略为直写，且写分配。若初始缓存为空，系统仅在对input和output的读写时会访问缓存。请在下表中填写执行上述程序，数组中每个元素的缓存命中情况。

	input array 列0	列1	列2	列3	output array 列0	列1	列2	列3
行0	miss	命中	miss	命中	miss	miss	miss	miss 1
行1	miss	命中	命中	miss 命中 2	miss	miss	miss	miss 2
行2	miss	miss	命中	miss 1	miss	miss	miss	miss 1
行3	miss	命中	miss	命中 2	miss	miss	miss	miss 2

16/4=4个 0x0=0 0x40=64

每一行1块 共2块

0x0

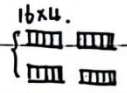
16. 考虑如下所示的C语言代码片段：

```
int input[2][128];
int sum=0;
for(int i=0;i<128;++i){
    sum+=input[0][i]*input[1][i];
}
```

假设系统中int变量为4字节，input数组的起始地址为0x0。系统仅在对input数组的读写时访问缓存，其它变量全部位于寄存器中。则：

1）如果存在一个512字节大小的缓存，该缓存两路组相联，块大小为16字节，使用LRU替换策略，缓存初始为空。则执行上述程序时缓存的命中率为多少？

$\frac{512}{2\times16}=16$ 组 $\frac{128}{4}$ 0～15　$\frac{16}{4}=4$ 个每个块存放4个数据



⓪ 0 1 2 … 127　$\frac{128}{4}=32$

① 0 … 127

程序访问 0，1 行的时 $N=128\times2$　$n=3\times16\times2\times2$

$R=\frac{128\times16}{2\times128\times8}=75\%$

2）在其它条件不变的情况下，增加缓存的总大小可以改善该程序的命中率？请说明理由。

不可以，因为块的第1个4字节int在初始一定为miss命中率上限为75%

3）在其它条件不变的情况下，增加缓存的块大小可以改善该程序的命中率吗？请说明理由。

可以，考虑极端情况块大小有有256字节则命中率为 $\frac{63}{64}>\frac{3}{4}$