A: 地址匹配控制. 控制该配置寄存器匹配物理地址的模式和范围.

5. (1) $\frac{2^{64}}{2^{12}} \times 8 = 2^{55}$ B

$= 32$ PB

(2) $\frac{2^{48}}{2^{12}} \times 8 = 2^{39}$ B

$= 512$ GB

(3) 将地址不同位段拆入不同级的表项管理, 减少了对高位地址映射信息的重复存储.

6. 由于空间局部性, 程序集中访问相邻地址. 若用高位做索引, 相邻地址高位基本一致, 冲突的可能性大大增加, 缓存无法发挥作用.

7. 程序通过虚拟地址访存, 因而也需从虚拟地址出发来读写缓存. 若映射位数相同, 则虚拟地址至缓存地址的转换简便, 提高访存速度.

8. 1) $1 \times 0.97 + 110 \times 0.03 = 4.27$

平均访问延时为 4.27 周期.

2) 缓存命中率 $P = \frac{64\text{kB}}{1\text{GB}} = \frac{1}{16384}$

$\therefore T = 1 \times \frac{1}{16384} + 110 \times \frac{16383}{16384} = 109.993$

平均访问延时为 109.993 周期

3) 若访存具有很好的空间局部性, 集中在某一小范围内基于连续访问, 则利用访问速度更快的缓存, 可大大提升处理器的访存性能. 但若在大范围内反复跳跃访问, 则因缓存缺失带来的额外开销反而会降低访存性能. 因此在程序设计时需考虑局部性原理以加快运行速度.

4) 设命中率 P. 若取得收益, 需:

$P \times 1 + (1-P) \times 110 < 105$

$P > 4.59\%$

9.

编号	组数量	索引位数	标记位数	偏移位数
1	32	5	21	6
2	8	3	23	6
3	1	0	26	6
4	256	8	18	6
5	64	6	19	7
6	256	8	18	6
7	64	6	20	6
8	32	5	20	7

10. 1) $0.22 + 100p1 < 0.52 + 100p2$

$p1 < p2 + 0.003$

$\therefore$ 当 $p1$ 比 $p2$ 高 0.3% 以内时, A 优于 B.

2) $0.22(1 + k p1) < 0.52(1 + k p2)$

当 $p1 < \frac{15}{11} + \frac{26}{11} p2$ 时, A 优于 B.

11. 直接映射：16个组，以16进制最低位为索引。
共请求2次0x1，5次0x5，且标签各不相同。
替换5次。

2路组相联：8组，每组2路，以索引地址低3位为索引。共请求2次001，5次101。前2次不替换，后续均需替换。
替换3次。

4路组相联：4组，每组4路，低2位为索引。
请求7次均为01，前4次不替换。
替换3次。

8路组相联：2组，每组8路，最低位索引
请求7次均为1，共8路，均不替换。
替换0次。

12. 顺序遍历96位数组100次，均进行一次写操作。
数组每位宽 $\frac{32}{8}$=4字节

A：首次加载 $\frac{256}{4}$=64次后缓存满，缺失 $\frac{256}{16}$=16次。
剩余32次加载恰好替换完一路，按LRU规则，
替换1~32位，缺失8次。
第二次遍历，按LRU原则，第二路被替换为1~32位，
此后第一路被替换为33~64位，第二路换为65~96位。
交替进行至100次遍历完毕。每32次访存均缺失8次。
∴A缺失率为 $\frac{8}{32}$=25%

B：首次加载64次后缓存满，缺失16次。
65~96位按直接映射原则，替换1~32位，缺失8次
此后遍历第二次，1~32位直接映射，替换65~96位，
但33~64位不被替换，全部命中。
第2~100次遍历，缓存前8组依次被1~32位
和65~96位占据，第9~16组固定被33~64位占
据，每次遍历，访存96次、缺失16次。
∴B缺失率为 $\frac{24+99\times16}{96\times100}$ = 16.75%。

13. for (int j=0; j<128; ++j){
for(int i=0; i<64; ++i)
A[j][i]=A[j][i]+1;
}
将i在内层循环，使访问地址更连续。

14. 假设数组为64×128位的二维数组，代码对数组完整遍历。
(1) int为4字节，每块可存8位，整个缓存可存1024位。
优化前：每读写一次，下次读写64位后的值。
前16次读写均位于不同块内，缺失16次。
此后偏移量相同，缓存均被替换。
∴每读写一次均缺失一次。
优化前缺失128×64=8192次。
优化后：顺序访问，每1024次读写缺失 $\frac{1024}{8}$=128次。
优化后缺失 $\frac{128\times64}{1024}\times128$=1024次。

(2) 优化前：缓存共128块，内层循环第一次刚好占满。此后依次读写块内的第2~8位，无缺失。有即可满足需求。全部缺失均为首次访问时的强制缺失。在块大小不变的情况下，增加缓存大小无法改善命中率。

外层遍历完该块后，需整体替换。

∴每读写1024次缺失128次，共缺失1024次。

优化后：顺序访问，每读写1024次缺失128次。

共缺失1024次。

(2)不能改善。本程序分两路顺序访存，2路缓

(3) 优化前：内循环一次加载128块，均会被重复访问8次。在直接映射下还需确保均被映射至不同位置。因此缓存容量应与数组容量相当。

共128×64×4 B = 32kB

优化后：连续访问，不重复读写同一块内的数据，故一块缓存即可。

需32B。

15.

	input				output			
	0	1	2	3	0	1	2	3
0	miss	miss	hit	miss	miss	miss	miss	miss
1	miss	hit	miss	hit	miss	miss	miss	miss
2	miss	miss	hit	miss	miss	miss	miss	miss
3	miss	hit	miss	hit	miss	miss	miss	miss

16. (1) 缓存共16组，每组2路

input[i][j]地址比input[i][j]高128×4字节，

因此每次存入同一组的不同路中。

此后各块顺序访问，完成后加载下两块。。。

每缓存4次缺失1次，

命中率为75%。