6 1. 地址局部性：最近被访问的数据内存地址很可能在不久再次被访问到，这种局部性使高位部分更有可能作为标签，因为它们通常会保持不变或以小幅度变化

2. 标签大小往往大于索引和偏移量，高位可以更好地区分内存多的存储块。如果将中间位作为标签，由于其包含信息较少，可能无法捕捉到

3. 中间位通常包含了不同的块号或页号，其变化范围较广，可有效减少冲突

7. 让页号部分与缓存标签部分具有相同位数，地址转换时，可以直接将虚拟页号直接映射到缓存的标签，不需

8. (1) $0.97\times1+0.03\times110=4.27$

(2) $1GB=1024MB=1024\times1024KB$

$1GB=2^{20}KB$

~~$\frac{2^6}{2^{20}}\times 0.03\times110+\frac{2^6}{2^{20}}\times$~~

$\frac{2^6}{2^{20}}\times4.27+\frac{2^{20}-2^6}{2^{20}}\times105\approx105$

(3) 局部性原理让机算机程序不随机访问数据的程序使存放相邻数据的缓存起到加速CPU访存的作用

(4). $110\cdot(1-x)+1\cdot x < 105$

$5 < 109x$

$4\times1024B$ ↑ $64B$ $x > 4.6\%$

9.

	缓存大小 KB	块大小 Byte	相联度	组数量	组索引位数	标签位数	偏移位数
1	32	4	64	2	32	5	21	6
			8	8	3	23	6
			全	1	0	26	6
	32	16	64	1	256	8	18	6
	32	16	128	2	64 (128)	6	19.	7
	32	64	64	4	256 (1024)	8	18	6
	32	64	64	16	64	6	20	6
	32	64	128	16	32	5	20.	7

(32)

(1) A. 8KB 直接映射

$T_A = 0.22(1-P_1) + P_1\times100$

$T_B = 0.52\times(1-P_2) + P_2\times100$

$T_A < T_B \Leftrightarrow (100-0.22)P_1 < 0.3 + (100-0.52)P_2$

$99.78P_1 < 0.3 + 99.48P_2$

(2) $0.22(1-P_1) + P_1\times0.22k < 0.52\times(1-P_2) + P_2\times0.52k$

11. 16个块 ~~块64B~~ 0×1007= 16+16^4×1

4 6 3

~~16×64×8=2^13~~ =

对16取余

~~真~~: 16个位置 直: ~~无块替换~~ 6次

~~0x1007~~= 2路: 对8取余 5次

4路 对4取余 3次

8路 对2取余 6次

~~24~~ 16

12. 块16B. Cache 256B

缓A 2路组相联 缓B 直接映射

对A. ~~8个位置~~ 8个组数. 16个块

256÷4=64个整数

96×4B=384B 会替换

内存 0 4 8 12

12. 对A 一个块16B, 可放4个整数

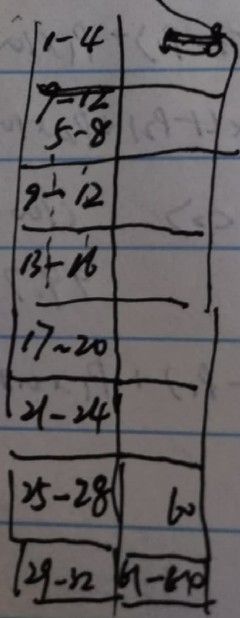
Cache有16个块, 故一共可放64个整数

A为2路组相联

共有96×100=9600次访问

96-64=32 ~~缺失率为古=25%~~

64×古



这些已失 64×古=25%

13. j在外，i在内

$8\sqrt{128}$ 16
8
48

14. (1) 2^{12} 4×1024B

2^5 32B 有 $2^7=128$ 个块 一个块8个整数

优化前 由于 A[0][0] A[0][1] …… A[0][7] 为一个块内
…… 8个
A[0][63] 为一个块内

A[1][0] A[1][1] …… A[1][7] 为一个块内

$\frac{1}{8}\times 64\times 128=1024$

A[127][0] A[127][1] …… A[127][63] 为一个块内

优化后 $\frac{1}{8}\times 64\times 128=1024$

(2). FIFO 替换 128 □□□……

优化前 $\frac{1}{8}\times 64\times 128=1024$

优化后 1024

(3)

Cache 32B
2个块

块	列0	1	2	3	列0	1	2	3
行0	miss	hit ✓	✓	✓	miss			
1	miss	✓	✓	✓	miss	miss		
2	miss	✓	✓	✓	miss			
3	miss	✓	✓	✓	miss			

6 (1) 未命中 $4\times128=256$ 总 $128\times2=256$

$256-256=0$

(2) 可以. 缓存大小↑, 可容纳更多的缓存块

从而减少缓存未命中的次数

(3) 增加块大小可能会增加缓存启动时间

但对连续访问命中率影响较小