

第五题上交作业已做

5.6. 若采用地址的高位进行索引，可能会使得地址相隔一个块大小的两个变量被映射到同一组中，产生错误，而若使用地址中间位索引，则可将不同元素分别映射到高速缓存组的不同组中，提高高速缓存利用率，也可增加命中率。

7. 使地址的组索引与块内偏移的各位数与虚拟内存页偏移位数相同，可使得虚拟地址的访问也可直接用来查找缓存，而缓存块内偏移量和组索引也可直接访问虚拟内存，实现TLB和缓存并行访问。同时，还可以减少内存冲突和内存管理开销，更好地利用局部性原理提高缓存命中率，来整体提高系统性能。

8. (1) $T_{delay} = 1 \times 0.97 + 110 \times 0.03 = 4.27$ (个周期)

(2) 由于是完全随机访问，故命中率近似认为 $\frac{64}{1 \times 1024 \times 16} \approx 0.006\%$ ，近似认为命中率为0。

因此， $T_{delay} = 1 \times 2^{-14} + (1 - 2^{-14}) \times 110 \approx 109.99$ (个周期)，故此时 $T_{delay} \approx 110$ (个周期)

(3) 在第1问中，程序的平均缓存缺失率为3%，说明了程序的局部性较好，故此时存储器及平均访问延时仅为4.27个周期。而在第2问中完全随机访问，故可认为几乎不存在局部性，此时的 T_{delay} 便需要110个周期。由此可见，有局部性且其较高的程序可以有效利用缓存，从而减少了访问内存的需要，可减少平均访问的延时，从而提高处理器的性能。

14) 命中率为x $x + 110(1-x) < 105 \quad 109x > 5 \quad x > 0.04587$

平均缓存命中率高于4.587%，一定能使存储器使用110时获得性能收益。

	扇区 地址位数(8bit)	缓存大小(kB)	块大小(BYTE)	相联及 淘汰量	组索引 位数	标签 (8bit)	偏移 (8bit)
1	32	4	64	2	32	5	21
2	32	4	64	8	8	3	23
3	32	4	64	全相联	1	0	26
4	32	16	64	1	256	8	18
5	32	16	128	2	64	6	19
6	32	64	64	4	256	8	18
7	32	64	64	16	64	6	20
8	32	64	128	16	32	5	20

10. (1) 由题意可知, $T_A = 0.22 + 100P_1$, $T_B = 0.52 + 100P_2 \therefore 0.22 + 100P_1 < 0.52 + 100P_2$

即 $100(P_1 - P_2) < 0.3 \therefore P_1 < P_2 + 0.003$ 时, 采样 A 平均内存访问时间优于采样 B.

(2) 由题意知, $T_A = 0.22 + 0.22kP_1$, $T_B = 0.52 + 0.52kP_2 \therefore 0.22 + 0.22kP_1 < 0.52 + 0.52kP_2$

$\therefore 0.22kP_1 < 0.3 + 0.52kP_2 \therefore P_1 < \frac{0.3}{0.22k} + \frac{0.52}{0.22}P_2$, 即 $P_1 < \frac{26}{11k}P_2 + \frac{15}{11k}$ 时, 采样 A 平均内存访问时间优于采样 B.

11. (1) 有 16 个块, 若为直接映射, 则需要 4 位地址索引 $\therefore 0x1001$ 与 $0x1021$, $0x0005$ 与 $0x1045$, $0x1455$ 与 $0x1305$.

$0x1305$ 与 $0x2005$, $0x2005$ 与 $0xff05$ 均会发生一次替换 \therefore 共发生 5 次替换.

(2) 2 路相联: $16 \div 2 = 8$ (组), 需要 3 位地址索引. 所以此时 $0x1001$ 和 $0x1021$ 映射到一组, $0x1005$ 与 $0x1045$ 映射到另一组而后者 $0x1305$, $0x2005$, $0xff05$ 均需发生一次替换 \therefore 一共发生 3 次替换.

(3) 4 路相联: $16 \div 4 = 4$ (组), 需要 2 位地址索引, 同理可知共需发生 3 次块替换.

(4) 8 路相联: $16 \div 8 = 2$ (组), 需要 1 位地址索引, 由于只有 2 个地址, 故可全部放下, 共发生 0 次替换.

12. $256 \div 16 = 16$ (个) 块, 故 A 为 8 组每组 2 块; B 为 16 组, 而缓存带宽为 $256 \div 4 = 64$ (字). 每块中放 4 个字
直接映射: 一块中有 4 个元素, 而一次访问涉及 96 个元素 故 32~64 位只会第一次查找对 miss, 后续均为 hit
但其他位的产生替换, 均为 miss \therefore 缺失率 = $\frac{64 \times 100 + 32}{96 \times 100} \times \frac{1}{4} = 16.75\%$.

2 路相联: \because 一共有 96 个元素多于 64 个来, 故会不断发生替换 (映射到后存命中则阻塞), 故即每一次访问所取的块
(每两个字)发生一次 miss. 缺失率 = $\frac{24 \times 100}{96 \times 100} = 25\% \therefore$ 在该例中 2 路相联缺失率高于直接映射

13. \because A[i] 与 A[i][j+1] 不相邻, 故应将 i 套为外层循环, j 为内层

代码: $\text{for } i=0; i<64; ++i \{$

$A[i][j] = A[i][j+1];$

}

14. (1) 全相联, 故当外层循环 8 次完后的情况, 后续的没有访问均不会发生 miss \therefore 优化前 miss 数 = $8192 \div 8 = 1024$ (次)

优化后的情况与 (1) 中相同, miss 数 = 1024 (次)

(2) 优化前: 需要每个 A[i][j] 不被重用, 故需 $128 \times 64 \times 4 = 32768$ Byte = 32 kB

优化后: 由于每次使用完元素后这个元素对后续就无影响了, 故仅需一个块即可, 故需 32 Byte 即可.

	input				output				
15.	列0	列1	列2	列3	列0	列1	列2	列3	由起始一个块中存放4个int值， 缓存之中一共有2个块。
行0	miss	hit	hit	hit	miss	miss	miss	miss	缓存之中一共有2个块。
行1	miss	hit	hit	hit	miss	miss	miss	miss	故在第0行中，第一次miss后便hit
行2	miss	hit	hit	hit	miss	miss	miss	miss	在第1行时，只会出现miss
行3	miss	hit	hit	hit	miss	miss	miss	miss	每时每次跨行，故全miss。

16. (1) $512 \div 16 = 32$ (个)块, 每个块中存放4个int型[0][i]与[0][i+4]是映射到同一组(差了128+4=32个块)

.. 每个块第一次访问时会miss, 后来不会. 命中率 = $\frac{128-32}{128} \times 100\% = 75\%$.

(2) 增加缓存大小可增加块数量, 从而增加命中率. 而该程序中一次循环恰好对应于块数32块,

不能 在该程序中无对于组复用的情况出现, 故增加缓存总大小不能改善程序的命中率.

(3) 增加块大小, 则增加同一个块中存放的int变量的数量, 而命中率不变, 但在该程序中

[0][0]与[0][i+4]块相同, 故在每次miss后, hit的int数会增加. 增加块大小能改善命中率.