May. 16th (Week 13) Chap. 4

6.

标签	索引	块内偏移
000…	000	000…
000…	001	000…

如上所示，使用中间位作为索引，则连续地址的索引不同，对应不同的组，同时存在缓存中。若高位作索引，连续地址的索引相同，对应同一组，读取下一项就需从缓存中替换此前的项。

7. 采用 VIPT 技术，用虚拟地址页内偏移中的几位作为组索引位。这样，在虚拟页号经 TLB 翻译为物理地址的同时，缓存索引与之并行，可减少命中时间。于是组索引位 + 块内偏移位 ≤ 页内偏移位数。通常取等号，否则浪费了地址位。

8. (1) $t = 1 \times 97\% + 110 \times 3\% = 4.27$ cycles

(2) 每次读取的命中概率为 $64\ KB / 1\ GB = 2^{-14}$

$t = 1 \times 2^{-14} + 110 \times (1 - 2^{-14}) \approx 110$ cycles. > 105 cycles

(3) 基于空间局部性和时间局部性，若处理器读取的数据与此前数据在空间、时间上相近，则可从缓存中取得，减少延时；对于完全随机读取的程序，局部性失去意义，由于缓存很小，几乎每次读取都出现缓存缺失，此时不经缓存直接访存反而更优。

(4) $h + 110 \times (1 - h) < 105 \Rightarrow h > 4.59\%$

平均缓存命中率需高于 4.59%，即有性能收益。

10. (1) $t_A < t_B$，即 $0.22 + 100 p_1 < 0.52 + 100 p_2 \Rightarrow p_1 < p_2 + 0.3\%$

(2) $t_A < t_B$ 即 $0.22 + p_1 k \times 0.22 < 0.52 + p_2 k \times 0.52 \Rightarrow 11 k p_1 < 26 k p_2 + 15$

9.

编号	地址bit	KB 缓存大小	Byte 块大小	相联度	组数	索引位	标签位	偏移位
1	32	4	64	2	32	5	1	6
2	32	4	64	8	8	3	3	6
3	32	4	64	全相联(64)	1	0	6	6
4	32	16	64	1	256	8	0	6
5	32	16	128	2	64	8	1	7
6	32	64	64	4	256	8	2	6
7	32	64	64	16	64	6	4	6
8	32	64	128	16	32	5	4	7

11.

	直接映射 mod 16	2路 mod 8	4路 mod 4	8路 mod 2
				→ 组数量
0x1001	1	1	1	1 → 索引
0x1005	5	5	1	1
0x1021	1	1	1	1
0x1045	5	5	1	1
0x1305	5	5	1	1
0x2ee5	5	5	1	1
0xff05	5	5	1	1
块替换次数：	7	3	3	0

12.

A：2路组相联 × 8组 × 16 Bytes

B：1路 × 16组 × 16 Bytes

int32_t 大小为 32 bit = 4 Bytes.

1块 含 4个 int32_t.

下标	块地址	索引A	索引B
0	0	0	0
4	1	1	1
…	…	…	…
32	8	0	8
36	9	1	9
…	…	…	…
92	23	7	7

May. 16th (Week 13) Chap. 4

6.

标签	索引	块内偏移
000···	000	000···
000···	001	000···

如上所示，使用中间位作为索引，则连续地址的索引不同，对应不同的组，同时存在缓存中。若高位作索引，连续地址的索引相同，对应同一组，读取下一项就需从缓存中替换此前的项。

7. 采用 VIPT 技术，用虚拟地址页内偏移中的几位作为组索引位。这样，在虚拟页号经 TLB 翻译为物理地址的同时，缓存索引与之并行，可减少命中时间。于是组索引位 + 块内偏移位 ≤ 页内偏移位数。通常取等号，否则浪费了地址位。

8. (1) $t = 1 \times 97\% + 110 \times 3\% = 4.27$ cycles

(2) 每次读取的命中概率为 $64\,KB / 1\,GB = 2^{-14}$

$t = 1 \times 2^{-14} + 110 \times (1 - 2^{-14}) \approx 110$ cycles. > 105 cycles

(3) 基于空间局部性和时间局部性，若处理器读取的数据与此前数据在空间、时间上相近，则可从缓存中取得减少延时；对于完全随机读取的程序，局部性失去意义，由于缓存很小，几乎每次读取都出现缓存缺失，此时不经缓存直接访存反而更优。

(4) $h + 110 \times (1 - h) < 105 \Rightarrow h > 4.59\%$

平均缓存命中率需高于 4.59%，即有性能收益。

10. (1) $t_A < t_B$，即 $0.22 + 100 p_1 < 0.52 + 100 p_2 \Rightarrow p_1 < p_2 + 0.3\%$

(2) $t_A < t_B$ 即 $0.22 + p_1 k \times 0.22 < 0.52 + p_2 k \times 0.52 \Rightarrow 11 k p_1 < 26 k p_2 + 15$

9.

编号	地址bit	大B 缓存大小	Byte 块大小	相联度	组数	索引位	标签位	偏移位
1	32	4	64	2	32	5	1	6
2	32	4	64	8	8	3	3	6
3	32	4	64	全相联(64)	1	0	6	6
4	32	16	64	1	256	8	0	6
5	32	16	128	2	64	8	1	7
6	32	64	64	4	256	8	2	6
7	32	64	64	16	64	6	4	6
8	32	64	128	16	32	5	4	7

11.

	直接映射 mod 16	2路 mod 8	4路 mod 4	8路 mod 2	→ 组数量
0x1001	1	1	1	1	→ 索引
0x1005	5	5	1	1	
0x1021	1	1	1	1	
0x1045	5	5	1	1	
0x1305	5	5	1	1	
0x2ee5	5	5	1	1	
0xff05	5	5	1	1	
块替换次数：	7	3	3	0	

12. A：2路组相联 × 8组 × 16 Bytes

B：1路 × 16组 × 16 Bytes

int32_t 大小为 32 bit = 4 Bytes

1块 含 4个 int32_t

下标	块地址	索引A	索引B
0	0	0	0
4	1	1	1
…	…	…	…
32	8	0	8
36	9	1	9
…	…	…	…
92	23	7	7

缓存B：直接映射。 $i=0$ 时，数组首次读入缓存，下标 ≥ 64 时需有块替换。

外循环 1 轮后，缓存中数据下标为：64～95，32～63。

每次循环需重读 0～31 和 64～95，而 32～63 始终在缓存中。

N很大时，缺失率为 $\frac{48}{96} \times \frac{1}{4} = \frac{1}{8}$

缓存A：2路组相联。缓存中已存入 0～31、32～63 后，以 64～96 替代 0～31（LRU），再以 0～31 替代 32～63，再以 32～63 替代 64～96…

N很大时，缺失率为 $\frac{1}{4}$。

13. 为使每次读（写）的数据在内存中相邻，只需交换循环次序，改为按行处理数组：

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

14. (1) 优化前：$64 \times 128 = 8192$ 次

优化后：$2^{13}/8 = 1024$ 次

(2) 优化前：$2^{13}/8 = 1024$ 次

优化后：$2^{13}/8 = 1024$ 次

(3) 优化前：需使每一块的块地址不同，即 1024 块，32 KB。（考虑需为2的幂）

优化后：总是读取相邻位置，只需 1 块，32B。

15.

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

16. (1) 16组 × 2路 × 16字节.

相邻4个数，仅第1个需访存. 命中率 $3/4 = 75\%$

(2) 不能。该条件下缓存总容量不是瓶颈，随着程序进行，组索引较小的容量占据缓存但不再被访问，已浪费容量。实际上，容量改为32B，命中率仍为75%.

(3) 可以。若块大小增大，一次可读入更多相邻数据，使用这些数据时不再访存.

如块大小增至32B. 命中率为 $7/8 = 87.5\%$