

S.11

(1) 如下图所示:

TLB					
Address	Virtual Page	TLB H/M	Valid	Tag	Physical Page
4669	1	TLB miss PT hit PF	1	11	12
			1	7	4
			1	3	6
			1 (last access 0)	1	13
2227	0	TLB miss PT hit	1 (last access 1)	0	5
			1	7	4
			1	3	6
			1 (last access 0)	1	13
13916	3	TLB hit	1 (last access 1)	0	5
			1	7	4
			1 (last access 2)	3	6
			1 (last access 0)	1	13
34587	8	TLB miss PT hit PF	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 0)	1	13
48870	11	TLB miss PT hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 4)	11	12
12608	3	TLB hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	11	12
49225	12	TLB miss PT miss	1 (last access 6)	12	15
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	11	12

(2) 如下图所示:

ILB					
Address	Virtual Page	TLB H/M	Valid	Tag	Physical Page
4669	0	TLB miss PT hit	1	11	12
			1	7	4
			1	3	6
			1 (last access 0)	0	5
2227	0	TLB hit	1	11	12
			1	7	4
			1	3	6
			1 (last access 1)	0	5
13916	0	TLB hit	1	11	12
			1	7	4
			1	3	6
			1 (last access 2)	0	5
34587	2	TLB miss PT hit PF	1 (last access 3)	2	13
			1	7	4
			1	3	6
			1 (last access 2)	0	5
48870	2	TLB hit	1 (last access 4)	2	13
			1	7	4
			1	3	6
			1 (last access 2)	0	5
12608	0	TLB hit	1 (last access 4)	2	13
			1	7	4
			1	3	6
			1 (last access 5)	0	5
49225	3	TLB hit	1 (last access 4)	2	13
			1	7	4
			1 (last access 6)	3	6
			1 (last access 5)	0	5

较大的 Page 大小可以降低 TLB 的利用率, 但会增加碎片, 也会降低物理内存利用率。

13) 两路组相联如下:

				TLB				
Address	Virtual		Index	TLB H/M				
	Page	Tag			Valid	Tag	Physical Page	Index
4669	1	0	1	TLB miss PT hit PF	1	11	12	0
					1	7	4	1
					1	3	6	0
					1 (last access 0)	0	13	1
					1 (last access 1)	0	5	0
2227	0	0	0	TLB miss PT hit	1	7	4	1
					1	3	6	0
					1 (last access 0)	0	13	1
					1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
13916	3	1	1	TLB miss PT hit	1	3	6	0
					1 (last access 0)	1	13	1
					1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
34587	8	4	0	TLB miss PT hit PF	1 (last access 0)	1	13	1
					1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
48870	11	5	1	TLB miss PT hit	1 (last access 0)	1	13	1
					1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
12608	3	1	1	TLB hit	1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
					1 (last access 5)	1	6	1
49225	12	6	0	TLB miss PT miss	1 (last access 6)	6	15	0
					1 (last access 5)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
					1 (last access 0)	0	5	0

14) 直接映射如下:

				TLB				
Address	Virtual		Index	TLB H/M				
	Page	Tag			Valid	Tag	Physical Page	Index
4669	1	0	1	TLB miss PT hit PF	1	11	12	0
					1	0	13	1
					1	3	6	2
					0	4	9	3
					1	0	5	0
2227	0	0	0	TLB miss PT hit	1	0	13	1
					1	3	6	2
					0	4	9	3
					1	0	5	0
					1	0	13	1
13916	3	0	3	TLB miss PT hit	1	3	6	2
					1	0	6	3
					1	2	14	0
					1	0	13	1
					1	3	6	2
34587	8	2	0	TLB miss PT hit PF	1	0	6	3
					1	0	6	3
					1	2	14	0
					1	0	13	1
					1	3	6	2
48870	11	2	3	TLB miss PT hit	1	0	6	3
					1	2	14	0
					1	0	13	1
					1	3	6	2
					1	2	12	3
12608	3	0	3	TLB miss PT hit	1	2	14	0
					1	0	13	1
					1	3	6	2
					1	0	6	3
					1	3	15	0
49225	12	3	0	TLB miss PT miss	1	0	13	1
					1	3	6	2
					1	0	6	3
					1	0	13	1
					1	0	6	3

所有的内存引用都必须对页表进行交叉引用, TLB 允许在不访问片外存储器的情况下进行。如果没有 TLB, 内存访问时间将显著增加

(5) 在两级方法中,  $2^{19}$  页条目被划分为 25 个段, 这些段是按需分配的。第二级表中的每一个都包含  $2^{19-8} = 2048$  个条目, 每个需要  $2048 \times 4 = 8\text{KB}$  覆盖  $2048$  个条目  $\times 8\text{KB} = 16\text{兆字节}$  ( $2^9$  的虚拟地址空间)

(6) 每块具有 2 个字的 16KB 直接映射高速缓冲存储器将具有 8 字节块, 因此  $16\text{KB}/8\text{字节} = 2048$  个块, 并且其索引字段将跨越地址位 13 到 3 (11 位索引, 1 位字偏移, 2 位字节偏移)。就像这样。高速缓存标签的标签 LSB 是地址位 14。

相反, 设计者需要使缓存具有双向关联性, 以将其大小增加到 16 KB。

S. 13. (1) 0 hits

(2) 1 hit

(3) 1 hits 或更少

(4) 1 hit. 地址序列正确, hit 的数量也正确

(5) 最好的区块是在未来造成最少失误的区块。不幸的是, 缓存控制器无法预知未来。我们最好的选择是做一个好的预测

(6) 如果知道一个地址具有有限的时间局限性, 并且会与高速缓存中的另一个块发生冲突, 它可以提高命中率

S. 15. (1)  $CPI = 1.5 + \frac{120}{10000} \times (15 + 175) = 3.78$

若 VMM 影响 doubles  $\rightarrow CPI = 1.5 + \frac{120}{10000} \times (15 + 350) = 5.88$

若 VMM 影响 halves  $\rightarrow CPI = 1.5 + \frac{120}{10000} \times (15 + 87.5) = 2.73$

(2) Nonvirtualized  $CPI = 1.5 + \frac{30}{10000} \times 1100 = 4.80$

Virtualized  $CPI = 1.5 + \frac{120}{10000} \times (15 + 175) + \frac{30}{10000} \times (1100 + 175) = 7.00$

Virtualized CPI with half I/O  $= 1.5 + \frac{120}{10000} \times (15 + 175) + \frac{15}{10000} \times (1100 + 175) = 5.69$

(3)

5.15.3 虚拟内存旨在为每个应用程序提供机器的整个地址空间的错觉。虚拟机的目的是为每个操作系统提供一种错觉，即整个机器都由它支配。因此，它们服务于非常相似的目标，并且提供了诸如提高安全性等好处。虚拟内存允许在同一内存空间中运行的多个应用程序不必管理它们的内存分离。

(4)

5.15.4 模拟不同的ISA需要对该ISA的API进行特定处理。每个ISA都有特定的行为，这些行为将发生在指令执行、中断、内核模式捕获等情况下。因此必须加以仿真。这可能需要执行比目标ISA中最初所需的多得多的指令来模拟每条指令。这可能会造成很大的性能影响，并使其难以与外部设备进行正常通信。如果可以动态地检查和优化仿真代码，则仿真系统可能比在其本机ISA上运行得更快。例如，如果底层机器的ISA有一条指令可以处理多个模拟系统指令的执行，那么执行的指令数量可能会减少。这与最近进行微操作融合的英特尔处理器的情况类似，允许用较少的指令来处理多个指令。

5.19 (1) srcIP and refTime fields. 2 misses per entry.

(2) Group the srcIP and refTime fields into a separate array.

(3) peak-hour (int status), // peak hours of a given status  
Group srcIP, refTime and status together

(4) 冲突命中不会发生在完全关联缓存中。  
通过从总命中率中减去强制未命中率和完全关  
联未命中率来计算容量未命中率

(5) Answers will vary depending on which data set is used.

(6) 例子: 4-block caches, direct-mapped vs. 2-way LRU.  
reference stream (blocks): 1 2 2 6)

6.6 (1) pk涉及的计算是:  $(m \times p \times n)$  乘法和  $(m \times p \times (n+1))$  加法  
在C语言中, 与单个元素相关的乘法和加法是依赖的。  
故加速比应该接近于4.

(2) 加速比将降低3倍, 这是为缓存丢失提供服务的成本的3倍。

(3) 方法是在C语言中通过跨列而不是跨行遍历矩阵来计算元素,