

ICS homework5

张运吉 211300063

7/12/2022

1 P3

(1)

$512\text{MB}/(64\text{M}\times 8\text{bit})=512\text{MB}/(64\text{M}\times 1\text{B})=8$ 所以需要 8 个 DRAM 芯片。

(2)

$2\text{GB}/512\text{MB}=4$, 需要 4 个内存条。

(3)

按字节编址, $4\text{GB} = 2^{32}\text{B}$, 所以主存地址一共有 32 位。

设主存地址为 $a_{31}\dots a_0$, 因为每个内存条是 512MB, 所以需要 29 位来指出内存条内的地址, 即 $a_{28}\dots a_0$, 因为是 $64\text{M}\times 8$ 位的芯片, 所以行列地址分别需要 13 位, 芯片内行地址位 $a_{28}\dots a_{16}$, 芯片内列地址位 $a_{15}\dots a_3$, $a_2a_1a_0$ 用于选择芯片。

2 P4

(1)

因为地址线有 16 位, 所以主存地址空间为 0000H FFFFH , 因为 0000H 7FFFH 分配给 ROM 区, 所以 RAM 区的地址范围是 8000H FFFFH , 需要 RAM

芯片 $32KB/(16K \times 4bit) = 4$ 个, 地址线中最高位即 A_{15} 用来区分 ROM 区和 RAM 区。

(2)

先计算新的 RAM 区的大小, ROM 区大小为 $2^{15}B = 32KB$, 主存空间总大小为 $16MB = 512 \times 32KB$, 所以 RAM 区大小为 $511 \times 32KB$, 需使用的芯片数量 $511 \times 32KB/(16K \times 4bit) = 2044$.

3 P5

数据块处理时间: $20000/500MHz = 0.04ms$

磁盘旋转一圈的时间: $1/(7200/60) = 0.00833s = 8.33ms$, 所以平均旋转等待时间为 $8.33/2 = 4.17ms$

一个数据块的平局读取或写回时间: $4KB/(40MB/s) = 0.1024ms$

所以数据块平局读取或写回时间: $2 + 10 + 4.17 + 0.1024 = 16.27ms$

因为数据块是随机存放的, 所以“读出-处理-写回”一次的时间: $16.27 \times 2 + 0.04 = 32.58ms$

每秒可以完成这样的操作次数: $1000/32.58 \approx 30$

4 P8

(1)

因为主存地址空间大小是 1GB, 所以主存地址位数一共有 30 位, 记为 $A_{29}...A_0$, cache 行数: $64KB/128B = 512$, 采取直接映射, 所以 cache 行号占 9 位, 每个主存块 128B, 块内地址 7 位, 主存地址标记位有 $30 - 9 - 7 = 14$ 位。

综上, 主存地址的高 14 位是标记位, 中间 9 位是行索引, 低 7 位是块内地址。

(2)

因为采用直接映射的方式, 所以不考虑替换算法, 也就没有用于替换的控制位, 采用直写方式, cache 中没有 dirty 位, 所以每个 cache 行包含 1

个有效位, 14 个标记位, 128B 数据, 所以, cache 总容量为 $512 \times (1 + 14 + 128 \times 8) \text{bit} = 519.5 \text{Kbit} \approx 65 \text{KB}$

5 P12

(1)

数组 x 和 y 都是按存放顺序访问, 所以空间局部性好, 由于每个数组只访问一次, 所以没有时间局部性。

无法推断命中率的高低, 因为没有给出 cache 的容量、块大小、映射方式等信息。

(2)

cache 一共有两行, 主存块为 16 字节, float 类型四个字节, 所以一个主存块可以包含 4 个数组元素, 数组 x 存放在 40H 开始的连续字节中, $x[0]-x[3]$ 在内存中属于第四块, $x[4]-x[7]$ 在第 5 块, $y[0]-y[3]$ 属于第六块, $y[4]-y[7]$ 属于第七块, 所以 $x[0]-x[3]$ 与 $y[0]-y[3]$ 总是映射到 cache 第 0 行, $x[4]-x[7]$ 与 $y[4]-y[7]$ 总是映射到 cache 第 1 行, 每个 $x[i]$ 和 $y[i]$ 总是命中到同一行, 互相淘汰对方, 命中率为 0。

(3)

cache 一共有四行, 分为两个组, $x[0]-x[1]$ 属于第 8 块, 依此类推, $y[6]-y[7]$ 存放在第 15 块, 因为每个组有两行, 所以 $x[i]$ 和 $y[i]$ 可以存放在同一组的不同行中, 对于每一块中的两个数组元素, 在访问第一个元素总是未命中, 访问第二个总是命中, 所以命中率是 50%。

(4)

若是将 x 定义为 $\text{float}[12]$, 则使得 y 从主存第七块开始存放, 那么 $x[i]$ 和 $y[i]$ 存放 cache 在不同行中, 不会相互淘汰, 对于每一块中的四个数组元素, 第一个总是未命中, 后面 3 个总是命中, 所以命中率为 75%。

6 P13

当 cache 数据区大小为 32B 时：

	col=0	col=1	col=2	col=3	col=0	col=1	col=2	col=3
row=0	0/miss	0/miss	0/hit	0/miss	0/miss	0/miss	0/miss	0/miss
row=1	1/miss	1/hit	1/miss	1/hit	1/miss	1/miss	1/miss	1/miss
row=2	0/miss	0/miss	0/hit	0/miss	0/miss	0/miss	0/miss	0/miss
row=3	1/miss	1/hit	1/miss	1/hit	1/miss	1/miss	1/miss	1/miss

当 cache 数据区大小为 128B 时：

	col=0	col=1	col=2	col=3	col=0	col=1	col=2	col=3
row=0	4/miss	4/hit	4/hit	4/hit	0/miss	0/hit	0/hit	0/hit
row=1	5/miss	5/hit	5/hit	5/hit	1/miss	1/hit	1/hit	1/hit
row=2	6/miss	6/hit	6/hit	6/hit	2/miss	2/hit	2/hit	2/hit
row=3	7/miss	7/hit	7/hit	7/hit	3/miss	3/hit	3/hit	3/hit

7 P19

(1)

直接映射，s=64：访存顺序为：a[0],a[64],a[0],a[64] ...

a[0] 和 a[64] 都是对应主存块的第一个元素，a[0] 和 a[64] 相差 256B，正好是 8 个主存块，因此这两个元素所在内存块会映射到 cache 同一行，每次都会发生冲突，所以缺失率为 100%。

(2)

直接映射，s=63：访存顺序为：a[0],a[63],a[126],a[0],a[63],a[126]...

a[63] 所在主存块的第一个元素是 a[56]，a[126] 所在主存块的第一个元素是 a[120]，a[56] 与 a[120] 正好相差 256B，即 8 个主存块，所以会映射到 cache 同一行，每次都会发生冲突，而 a[0] 则不会发生冲突，所以缺失率是 67%。

(3)

2 路组相联, $s=64$: 访存顺序为: $a[0], a[64], a[0], a[64] \dots$

$a[0]$ 和 $a[64]$ 都是对应主存块的第一个元素, $a[0]$ 和 $a[64]$ 相差 256B, 正好是 8 个主存块, 因此这两个元素所在内存块会映射到 cache 同一组, 但由于是两路组相联, 所以可以放在同一组的不同行中, 只有第一次访问时会缺失, 后续访问不会发生冲突, 冲突率几乎是 0。

(4)

2 路组相联, $s=63$: 访存顺序为: $a[0], a[63], a[126], a[0], a[63], a[126] \dots$

$a[63]$ 所在主存块的第一个元素是 $a[56]$, $a[126]$ 所在主存块的第一个元素是 $a[120]$, $a[56]$ 与 $a[120]$ 正好相差 256B, 即 8 个主存块, 所以会映射到 cache 同一组, 但由于是两路组相联, 所以可以放在同一组的不同行中, 只有第一次访问时会缺失, 后续访问不会发生冲突, $a[0]$ 则存放在别的组中, 不会发生冲突, 总的来说, 只有第一次访问时会发生冲突, 后续访问不会发生冲突, 冲突率几乎是 0。

8 P20

由于 $2^{36}B/16KB = 2^{22}$, 物理页号的位数至少为 22, 每个页表项包括有效位、保护位、修改位、使用位、物理页号等, 所以一个页表项至少为: $22 + 4 = 26$, 虚拟页数为: $2^{40}B/16KB = 2^{26}$, 为了简化对页表的访问, 页表项的大小应是 2 的幂次, 这里我们取 32, 所以每个进程的页表大小为: $2^{26} \times 32bit = 256MB$.

问题: 构建出的页表太大, 导致无法一次性加入到内存中。

9 P21

(1)

低 7 位是页内偏移量, 高 9 位是虚页号, 虚页号中高 7 位是 TLB 标记, 低 2 位是 TLB 组索引。

(2)

低 7 位是页内偏移量，高 5 位是物理页号。

(3)

低 2 位是块内地址，中间 4 位是 cache 行索引，高 6 位是标记。

(4)

虚拟地址:067AH=0000 0110 0111 1010B, 所以虚页号为: 000001100B, 低 2 位是 00B, 所以映射到 TLB 中第 0 组, 高 7 位标记是 0000011B=03H, TLB 中对应的表项有效位是 0, 所以 TLB 缺失, 需要到内存的页表中查找, 查找 b 图中的 00C 处表项, 有效位是 1, 说明对应的页在内存中, 取出物理页号 19H, 和虚拟地址中的页内偏移组合形成物理地址: 110011 1110 10B, 去除物理地址中的 cache 行索引字段 1110, 直接找到 cache 中第 E 行, 其有效位是 1, 且标记 33H=110011B, cache 命中。根据物理地址的低 2 位 10B, 取出字节 2 中的内容 2D4AH。

10 P23

(1)

```
movl  $0,%ecx
.loop
cmpl  %ebx,%ecx
jge   .done
addl  (%edx,%ecx,4),%eax
incl  %ecx
jmp   .loop
.done
```

(2)

执行到程序 P 时, 处于保护模式并且采用分页虚拟管理方式, 所以 $PE = 1, PG = 1$.

(3)

IA-32 中 long 时 32 位带符号整数,所以指令 I 为 *addl (%edx,%ecx,4),%eax*, 寻址方式是” 基址 + 比例变址 + 偏移量”。

(4)

线性地址: $0x0 + 0x8048c08 = 0x8048c08$.

指令 I 的地址低 12 位是页内偏移量: 1100 0000 1000, 高 20 位是虚页号, 虚页号是: 0000 1000 0000 0100 1000.

虚页号中低 10 位是页表索引: 00 0100 1000, 高 10 位是页目录索引: 0000 1000 00.

指令 I 在内存地址为: $0x1020000 + 0xC08 = 0x1020C08$ 起的 4 个字节。

$P=1, R/W=0, U/S=1, A=1, D=0$.

(5)

取指令不会发生缺页异常, 因为指令 I 不在一个页的起始处, 在执行指令 I 前面的指令发生缺页时, 会将指令 I 一起调入内存; 但在取操作数 $a[0]$ 的时候可能会发生缺页异常, 因为 $a[0]$ 所在地址对应的页可能还没有加载到内存, 若是这样, 页故障线性地址是 $0x8049000$, 该地址保存在控制寄存器 CR2 中。

(6)

同 (5), 取指令时不会发生 TLB 缺失, 但取操作数可能会发生 TLB 缺失。20 为虚拟页号中高 18 位 TLB 标记, 低 2 位为 TLB 索引。

根据 TLB 标记和 TLB 索引, 在 TLB 中能找到一样且有效位为 1 的页表项, TLB 命中, 取出页框号 028B0H, 得到主存地址 $0x28b0c08$.

(7)

指令 cache 一共有 $8KB/32B = 256$ 行, 组数为 128。指令 I 的线性地址为 $0x8048c08$, 其中第 12 位为页内偏移量, 组索引为 1100000, 块内地址为 01000, 因为指令 I 不在某个主存块的起始位置, 因此在第一次执行指令 I 时, 不会发生 cache 缺失。

映射到 1100000 组中。

(8)

$N=2000$, 数组 a 的大小为 8000B, 链接后 a 的首地址为 0x8049300, 对应段基址为 0, 所以 a 的线性地址为 0x8049300, 这不是一个页面的起始地址, $8000/4K \approx 1.9$, 最后占三个页面。虚页号分别是 0000 1000 0000 0100 1001 和 0000 1000 0000 0100 1010 和 0000 1000 0000 0100 1011.

因为 $4 \times 1200 = 4800 > 4096$, 所以 $a[1200]$ 在第二个页面。