

4. RISC-V 物理内存保护 (PMP) 和页表硬件
有 2^{64} 个不同的虚拟地址，需要 $2^{64}/512 = 2^{55}$ bytes。
线程为特定的物理内存区域指定访问权限，其

配置寄存器如下形式：① PMP 控制寄存器中的
X/W/R 位有什么作用？

② 仅使用 48 位虚拟地址空间：

$$2^{48} \text{ bytes} / 4KB \times 8 \text{ bytes} = 2^{39} \text{ bytes}$$

③ PMP 配置寄存器中的 L 和 A 位作用？

解：① 空间利用率：只为实际使用的页表条目分配
权限，提高了对物理内存区域的额外保护。内存。

② 层次结构：通过分割虚拟地址空间为更小
的区域。

可覆盖页表中的权限设置。当发生内存访
问时，同时检查。

③ 精准加载：只在需要访问特定页时才会加载

④ 0L 位：lock，用于锁定 PMP 配置寄存器。
防止对其进行非特权级的写入操作。为 1 时，

允许进一步的配置更改，有助于确保配置
的完整性和平凡性。

⑤ Addressing：用于指示 PMP 自己寄存器的
地址范围模式。为 0，以字节为单位；为 1，

以对齐的四字节 (32 位) 区域进行编码。这样可
提供更大的地址范围，并减少 PMP 配置寄存器
的数量需求。

可灵活配置 PMP 机制来保护物理内存，并限
制对特定区域的访问权限。

5. 页大小 4KB，每页是条目使用 8 字节空间；内存
系统按字节寻址，使用完整的 64 位虚拟地址
时，一个单级页表需要多大的空间用于存储
页表：

由 每页 4KB，8 字节 = 512 个页表条目的地址。

嵌入式第十二次作业

Date.

6. 试简要分析为什么缓存一般使用地址中间位作为组索引、高位作为标签？
答：为了提高缓存的命中率和访问性能，这种方式被称为“组相联”缓存。

1. 缓存的组索引：缓存通常被分为多个组，中间位提供了足够位数，灵活、组相关，增加命中率。当两个或更多数据块具有相同的组索引时，可以在同一组中共享相同缓存行。

2. 缓存的标签：用标识数据块高位，可提供更大的地址空间，可避免与虚拟内存解：从缓存，平均缓存缺失率为3%，平均成为页表的标识冲突。唯一标识，以便进行正确的查找。

3. 命中率和性能：较小的硬件开销下提高了缓存的命中率，减少了因冲突的可能性，提高命中率，可以从缓存中直接获取，而不需要访存主有足以放下完整数据， L_1 大小为64KB，平均访问延时：

$$\text{命中率 } R = \frac{n}{N} = \frac{64\text{KB}}{1\text{GB}} = \frac{2^6 \times 2^{10}\text{B}}{2^{30} \times 2^{10} \times 2^{10}\text{B}} = 2^{-14}$$

$$\therefore T = t + \mu P = t + (1-R)P$$

$$= 1 + (1 - 2^{-14}) \times 109 = 110 \text{ cycle.}$$

7. 真实的：存储地址由组索引和块内偏移的每一位数与虚拟内存系统的页偏移位数

相同，如：

答：1. 简化地址转换：通过将地址组索引和块内偏移每一位数与页偏移每一位数保持一致，可简化地址转换的逻辑。有助于提高相同的位进行地址转换。有助于提高访问速度和降低延迟。

3. 利用以上结果，局部性原理如何影响处理器的访存性能？

答：缓存命中时延为1cycle，因为cache访问局部性使处理器能够在cache中直接读取数据。a情况：缓存命中率高，充分利用时间局部性佳延时小；缓存缺失延时为110cycle，

Date.

是由于局部性原理中的空间局部性引起，命中率
需要处理量额外付出到下一级cache。b：块长
有限而访存数据块范围又过大，未得到发挥。
缺块率较高，命中率低。

14) 至均命中率需要满足什么条件，才能使命中
率用L1时获得性能收益：

$$T = t + (1-R)P \leq 1.05$$

$$\Rightarrow P > 1 - \frac{1.05}{1.09} = \frac{5}{109}$$

即命中率至少达到 $\frac{5}{109}$ 才有性能收益。

9. 缺	地址位数 Bit	缓存大小 KB	块大小 Byte	相联度	组数量	回写策略	粒度 Bit	修改 Bit	偏移 Bit
1	32	4	64	2	32	5	21	6	
2	32	4	64	8	8	3	23	6	
3	32	4	64	全相联	1	0	26	6	
4	32	16	64	1	256	2	18	6	
5	32	16	128	2	64	6	19	7	
6	32	64	64	4	256	8	18	6	
7	32	64	64	16	16	6	20	6	
8	32	64	128	16	32	5	20	7	

10. β : 8KB直接映射缓存命中延时 0.22ns. $T_1 < T_2$, by $P_1 < P_2 + 0.003$.
缓存缺率为 P_1 ; β : 64组四路相联。假设直接映射代价分别是各条指令命中延时的 k 倍，
缓存命中延时 0.52ns, 缓存缺率为 P_2 . β 优于 β ?

假设直接映射代价为常数 100ns. β 优于 β ? $T_1 = t_1 + m_1 p_1 = t_1 + k t_1 p_1 = t_1 (1 + k p_1)$

解: $T_1 = t_1 + m_1 p_1 = 0.22 + 100 p_1$ $T_2 = t_2 + m_2 p_2 = t_2 + k t_2 p_2 = t_2 (1 + k p_2)$

$T_2 = t_2 + m_2 p_2 = 0.52 + 100 p_2$ $11 p_1 < \frac{15}{k} + 26 p_2$.

11. 16块，块大小为64字节。请求多块地址为： $0x1005 \equiv 1 \pmod{2}$

$0x1001, 0x1005, 0x1021, 0x1045, 0x1345, 0x1021 \equiv 1 \pmod{2}$

$0x2005, 0xffff05$ 。直接映射、2路组相联，共发生0次替换。

4~8~。缓存发生块替换的次数。

解：1) 直接映射：16组 1路。

$$0x1001: 1 + (2^4)^3 = 2^{12} + 1$$

$$0x1345: 2^{12} + 2^8 \cdot 3 + 5$$

$$0x1021: 5 + 2^{12}$$

$$0x1045: 2^{12} + 2^8 + 5$$

$$0x1005: 2^{12} + 2^5 + 1$$

$$0x1005: 2^3 + 14 \cdot 2^8 + 14 \cdot 2^4 + 2$$

$$0xffff05: 2^{12} \cdot 15 + 2^8 \cdot 15 + 5$$

$$0x1001 \equiv 1 \pmod{16}$$

$$0x1005 \equiv 5 \pmod{16} \dots \text{共发生4次替换}$$

② 2路组相联：2路8组

$$0x1001 \equiv 1 \pmod{8}$$

$$0x1005 \equiv 5 \pmod{8}$$

$$0x1021 \equiv 1 \pmod{8} \dots$$

共发生2次替换。

③ 4路组相联：4路4组

$$0x1001 \equiv 1 \pmod{4}$$

$$0x1005 \equiv 1 \pmod{4}$$

$$0x1021 \equiv 1 \pmod{4} \dots$$

共发生2次替换。

④ 8路组相联：8路2组

$$0x1001 \equiv 1 \pmod{2} \dots$$

12. 两个块大小均为16字节，总容量均为256字节的缓存：

A：两路组相联，防直接。LRU。运行下述程序时使用以上缓存的缺失率。

解：共 $256 \div 16 = 16$ 个块，一个块4行int

缓存A：2路8组

B：1路16组

总访问次数为 $96 \times 100 = 9600$ 次。

A 缺失次数：

$$96 \times \frac{1}{4} \times 100 = 2400 \text{ 次，缺失率 } 25\%.$$

B 缺失次数：

$$(64 + 32 \times 99) \times \frac{1}{4} = 16 + 8 \times 99 = 16 + 792 = 808 \text{ 次}$$

$$\text{缺失率为 } \frac{101}{1200} = 8.42\%.$$

13. A[i][j]与A[i][j+1]相邻。从利用缓存空间利用率角度，优化以下代码。

解：for (int j=0; j<128; j++) {

 for (int i=0; i<64; i++) {

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

}

14. 程序对数组A发生读写时访问缓存，其他变量全部位于寄存器中。

由 4KB 直接~，块大小 32 字节。优化前后代码
缓存缺失次数分别多少？

$$\text{解: } 4KB \div 32B = 128 \text{ 块}$$

$$128 \text{ 块中有 } 32 \div 4 = 8 \text{ 个 int 数据}$$

优化前：缺失次数为 $128 \times 6 = 768$ 次

$$\text{优化后: } 128 \times \frac{1}{8} \times 14 = 1024 \text{ 次.}$$

(2) 4KB 分相联，块大小 72 字节，FIFO 替换策略。

$$\text{解: 前: 缺失次数为 } 128 \times 64 \div 8 = 1024 \text{ 次.}$$

$$\text{后: 缺失次数为 } 128 \times \frac{1}{8} \times 64 = 1024 \text{ 次}$$

16. 小存在一个 512 字节大小的缓存，4KB 命中率

$$\text{解: 缺失次数 } N = 2 \times 128 = 256 \text{ 次}$$

$$\text{命中次数 } n = 16 \times 3 \times 2 \times 2 = 192 \text{ 次}$$

$$\text{命中率 } R = \frac{n}{N} = \frac{3}{4}$$

(2) 增加缓存页大小可以改善该程序命中率吗？

Cache 页大小增大：不能

(3) 增加缓存块大小？

能，增大 $\frac{\text{块大小} \div 4B - 1}{\text{块大小} \div 4B}$

(3) 块大小为 32 字节的直接映射缓存，除了必要的强制缺页外，而没有带多大缓存容量。

$$\text{解: } 128 \times 8 \times 32B = 32KB$$

$$\text{总: } 1 \times 32B = 32B$$

15. 填写数组中每个元素的缓存命中情况。

input 数组

output 数组

$3y_0 \ 3y_1 \ 3y_2 \ 3y_3$

$3y_0 \ 3y_1 \ 3y_2 \ 3y_3$

行 1 miss m h miss

m m m m

行 2 m m h m

m m m m

行 3 m h m h

m m m m