奔腾和ARM处理器采用多级页表，如何组织多级页表？多级页表如何进行地址转换？为了解决多级页表的速度问题，设置了TLB，说明使用TLB前后的访问时间。

Pentium

一、背景知识

Pentium，奔腾处理器，是英特尔公司开发的微处理器系列的一款处理器。Pentium作为英特尔 80486[微处理器](https://www.britannica.com/technology/microprocessor)的后继产品，于 1993 年推出，在单个[芯片](https://www.britannica.com/technology/integrated-circuit)上包含两个处理器和大约 330 万个[晶体管](https://www.britannica.com/technology/transistor)。采用 CISC（复杂指令集计算机）架构。

Pentium处理器的主要特点是 32[位](https://www.britannica.com/technology/bit-communications)地址总线、64 位数据总线、内置浮点和内存管理单元以及两个 8 KB[缓存](https://www.merriam-webster.com/dictionary/caches)。它的处理器速度范围为 60 到 200 兆赫 (MHz)。Pentium 很快成为个人电脑的首选处理器。当然，现在有更快、更强大的处理器出现，比如Pentium Pro (1995)、Pentium II (1997)、Pentium III (1999) 和 Pentium 4 (2000)。2006 年，Intel 推出了 Core 系列微处理器，Pentium 系列成为用于廉价个人计算机的中端产品线。

二、如何组织多级页表？

1. 分页管理的基本思想

分页管理的基本思想是将4GB的线性地址空间分成固定大小的页面。这些页面可以映射到物理地址空间和磁盘存储器上。

2. 分页管理机制使用的四种数据结构

1. 页目录

最多由1024个页目录项组成，每个页目录项为32位。页目录可包含在一个4KB的页面中。

1. 页表

最多由1024个页表项组成，每个页表项为32位。页表可包含在一个4KB的页面中。

1. 页面

指一个大小固定的、连续的、未分段的平面地址空间。

1. 页目录指针表

由4个64位项组成，每项指向一个页目录。该数据结构只在允许物理地址扩展时使用。

3. Pentium分页机制的组成

1. 页目录基址寄存器 PDBR
2. 一个页面目录
3. 至少一个页表

4. 分页机制

在奔腾处理器中，可允许的页面大小为4KB或者4MB。

对于4KB大小的页面，奔腾处理器使用两级分页的技术。首先页目录基址寄存器PDBR是CR3，指向页目录的基址。页面目录的每个条目PDE指向4MB内存的页表，同时PDE包含控制信息和指向页表的指针。与PDE一样，页表中每个条目PTE都指向一个4KB的页面，而且包含相应的控制信息。

补充：对于4MB大小的页面，为单级页表，相比4KB，取消了页表的查找，页面目录的每个条目PDE直接指向4MB内存。因此对于页面的索引从12位增加到22位，才能实现对4 MB 大小的页面的直接控制；而页目录基址寄存器PDBR中的指针减少为10位，直接指向内存的4MB页面。

而如何判断当前页面大小需要通过一个标志。页目录中的一项是页大小标志，如果设置了该标志，则表明页面的大小为 4 MB 而不是4 KB。如果设置了该标志，则页目录直接指向4MB页面，而绕过内部页表。

三、多级页表如何进行地址转换？

1. 页面大小为4KB时的两级页表地址转换过程

对于线性地址，通过使用页表中的20位指针与线性地址的低12位的组合，形成32位的物理地址。

为了实现地址转换，将线性地址分为3个部分：

* 高10位：页目录的索引
* 中间10位：页表的索引
* 低12位：页面的索引

具体的地址转换过程是：CPU 首先获得线性地址并以控制寄存器CR3（页目录基址寄存器PDBR）中的高20位作为页目录基地址的高20位，隐含的低12位为全0，获得32位的页目录基地址；再以线性地址的高10位页目录索引（×4）查找页目录，由相应的页目录项得到相应页表的基址；接着以线性地址的中间10位页表索引（×4）查找页表，由相应的页表项得到相应的页框基地址，将该页框基地址加上线性地址中的低12位页内偏移量得到最终的32位物理地址。

如下图：

图示, 工程绘图

描述已自动生成

图-4KB页面大小的地址转换

2. 页面大小为4MB时的地址转换过程

4 MB 页面的线性地址转换为物理地址的方式与 4 KB 页面非常相似。但是，省略了对页表的访问。

实现地址访问：

* 高10位：页目录的索引（指向 4 MB 内存页面的 10 位指针）
* 低22位：页面的索引

具体的地址转换过程为：当控制寄存器CR3（含一组允许结构扩展标志的寄存器）中页面规模扩展标志（PSE）置位，且页目录项（PDE）中的页面规模（PS）标志置位时，页面规模大小选定为4MB。此时线性地址对4MB页面的映射直接由页目录项完成，不经过页表。

如下图：

图示

描述已自动生成

图-4MB页面大小的地址转换

四、设置TLB以及使用TLB前后的访问时间

根据1995年版的奔腾用户手册，奔腾有一个代码TLB和两个数据TLB。

数据 TLB 由用于 4 KB 页面转换的 64 条目 TLB 和用于 4 MB 页面转换的 8 条目 TLB 组成。代码 TLB 是单个 32 条目 TLB，由 4KB 和 4MB 页面转换共享。4 MB 代码页以 4 KB 的倍数缓存。

首先对于一级页表，假设查页表找页表项和查地址找数据拼接地址的时间都分别为t。

**没有快表：**EAT=2t

**有快表**：假设查找快表需要的时间为λ

EAT = 【快表中找到】a\*λ +【快表没有找到】(1-a)\*(t【在内存中找】+λ【更新快表】)+t【转换为物理地址】。

对于4KB大小的页面，为两级页表。将页表划分为2级，因此需要访问内存3次，而系统存在快表且在快表命中时，只需要1次访存。具体计算如下：

**没有快表：**EAT=t（访问页目录）+t（访问页表）+t（物理地址转换）=3t

**有快表：**假设查找快表需要的时间λ

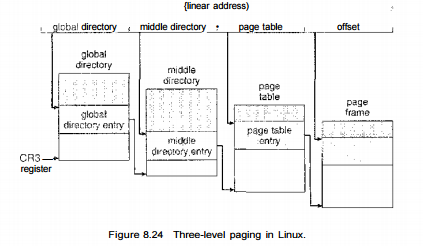
EAT = 【快表中找到】a\*λ +【快表没有找到】(1-a)\*(2t【在内存中找（页目录和页表）】+λ【更新快表】)+t【转换为物理地址】

五、扩展

尽管 Pentium 使用两级分页模型，但 Linux 被设计为在各种硬件平台上运行，其中许多是 64 位平台，在这些平台上两级分页是不合理的。因此，Linux 采用了三层分页策略，适用于 32 位和 64 位架构。Linux中的线性地址分为以下四部分：

|  |  |  |  |
| --- | --- | --- | --- |
| global directory | middle directory | page table | offset |

主要的地址转换过程如下：



六、参考文献

[1]许精明.保护模式下Pentium微处理器的存储管理方案[J].微机发展,2003(03):10-11+95.

[2]夏春梅.分页存储管理系统中内存有效访问时间的计算[J].滨州学院学报,2015,31(06):94-96.DOI:10.13486/j.cnki.1673-2618.2015.06.018.

[3] Improved Address-Space Switching on Pentium Processors by Transparently Multiplexing User Address Spaces--Jochen Liedtke

[4] intelpaging—protected-mode memory management

[5] UKEssays. November 2018. Pentium Memory Management Unit Computer Science Essay. [online]. Available from: https://www.ukessays.com/essays/computer-science/pentium-memory-management-unit-computer-science-essay.php?vref=1 [Accessed 28 December 2021].

[6] <https://padakuu.com/article/136-example:-the-intel-pentium>

ARM

一、背景知识

ARM是Advanced RISC Machine的缩写，即进阶精简指令集机器。arm更早称为Acorn RISC Machine，是一个32位精简指令集（RISC）处理器架构。也有基于ARM设计的派生产品，主要产品包括Marvell的XScale架构和和德州仪器的OMAP系列。ARM家族中32位嵌入式处理器占比达75%，由于ARM的低功耗特性，被广泛反应于移动通信领域、便携式设备等领域。

1983年Acorn电脑公司（Acorn Computers Ltd）开始开发一颗主要用于路由器的Conexant ARM处理器，由Roger Wilson和Steve Furber带领团队，着手开发一种新架构，类似进阶的MOS Technology 6502处理器。Acorn有一大堆建构在6502架构上的电脑。该团队在1985年时开发出ARM1 Sample版，并于次年量产了ARM2，ARM2具有32位的数据总线、26位的寻址空间，并提供64 Mbyte的寻址范围与16个32-bit的暂存器。

在1980年代晚期，苹果电脑开始与Acorn合作开发新版的ARM核心。1990年将设计团队另组成一间名为安谋国际科技（Advanced RISC Machines Ltd.）的新公司,。1991年首版ARM6出样，然后苹果电脑使用ARM6架构的ARM 610来当作他们Apple Newton PDA的基础。在1994年，Acorn使用ARM 610做为他们Risc PC电脑内的CPU。

ARM是一家微处理器行业的知名企业，该企业设计了大量高性能、廉价、耗能低的RISC （精简指令集）处理器，它只设计芯片而不生产。ARM的经营模式在于出售其知识产权核（IP core），将技术授权给世界上许多著名的半导体、软件和OEM厂商，并提供技术服务。

ARM的版本分为两类，一个是内核版本，一个处理器版本。内核版本也就是ARM架构，如ARMv1、ARMv2、ARMv3、ARMv4、ARMv5、ARMv6、ARMv7、ARMv8等。处理器版本也就是ARM处理器，如ARM1、ARM9、ARM11、ARM Cortex-A（A7、A9、A15），ARM Cortex-M（M1、M3、M4）、ARM Cortex-R，这个也是我们通常意义上所指的ARM版本。

二、如何组织多级页表？

1. 概念简述

在ARM系统中：

存储管理单元MMU功能：虚拟存储器空间到物理存储空间的映射。

系统控制协处理器CP15的寄存器C2用来保存页表的基地址。

页表：位于内存中的表。表的每一行对应于虚拟存储空间的一个页。

2. 多级页表概述

ARM MMU硬件采用2级页表结构：一级页表（L1）和二级页表（L2）。

一级页表只有一个L1主页表。L1主页表包含2中类型的页表项：保存指向二级页表起始地址指针的页表项和保存用户转换1MB页的页表项。L1主页表也称为段页表。

L1主页表将4GB的地址空间划分为多个1MB的段，L1主页表是一个混合表，可作为L2页表的页目录，也可作为用于转换1MB虚拟页（也可称为一段）的普通页表。当L1页表作为页目录时，其页表项包含的是代表1MB虚拟空间的L2粗页或L2细页表的指针；当L1页表用户转换一个1MB的段时，其页表项包含的是未来路存储器中1MB页帧的时候地址。

2. 一级页表项

一级页表支持4种类型的页表项: 1MB段转换项;指向L2细页表的目录项;指向L2粗页表的目录项;产生中止异常的错误项。

系统通过页表项的最低2位 [1:0] 来确定页表项的类型。页表项的格式要求L2页表的地址必须与其页大小的倍数对齐。L1页表的各种页表项格式如图2-1所示。

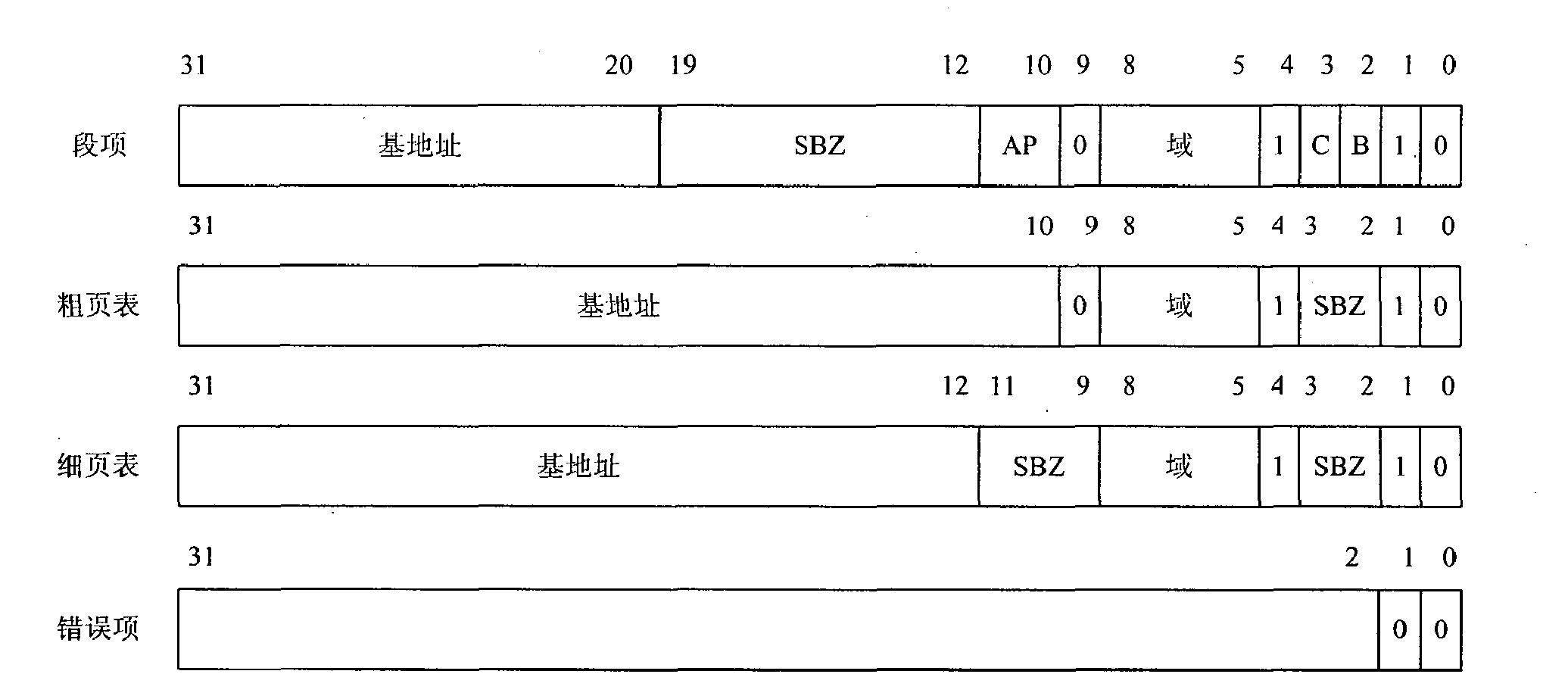


图2-1 L1页表项

3. 二级页表项

L2页表有4种可能的页表项:定义64KB页帧属性的大(large) 页表项;定义4KB页帧的小(small)页表项;定义1KB页帧的极小(tiny)页表项;访问时产生页错误中止异常的错误页表项。

L2页表的页表项格式如图2-2所示。MMU通过页表项的最低2位来确定L2页表项的类型。

大页表项包含一个64KB物理存储块的基地址，同时它还含有4组权限位域,以及页的cache和写缓冲器属性。每一组访问权限位域代表虚存页的1/4， 这些页表项可以看成是16KB子页，以更好地控制64KB页的访问权限。

小页表项保存一个4KB物理存储块的基地址，同时它也含有4组权限位域，以及页的cache和缓冲器属性。每一组访问权限位域代表虚存页的1/4，这些页表项可以看成是1KB子页，以更好地控制4KB页的访问权限。

极小页表项提供一个1KB物理存储块的基地址，同时它含有一个访问权限位域，以及页的cache和缓冲器属性。

错误页表项产生存储页访问错误。错误条件会导致预取指中止或数据中止，这取决于具体的存储器访问类型。

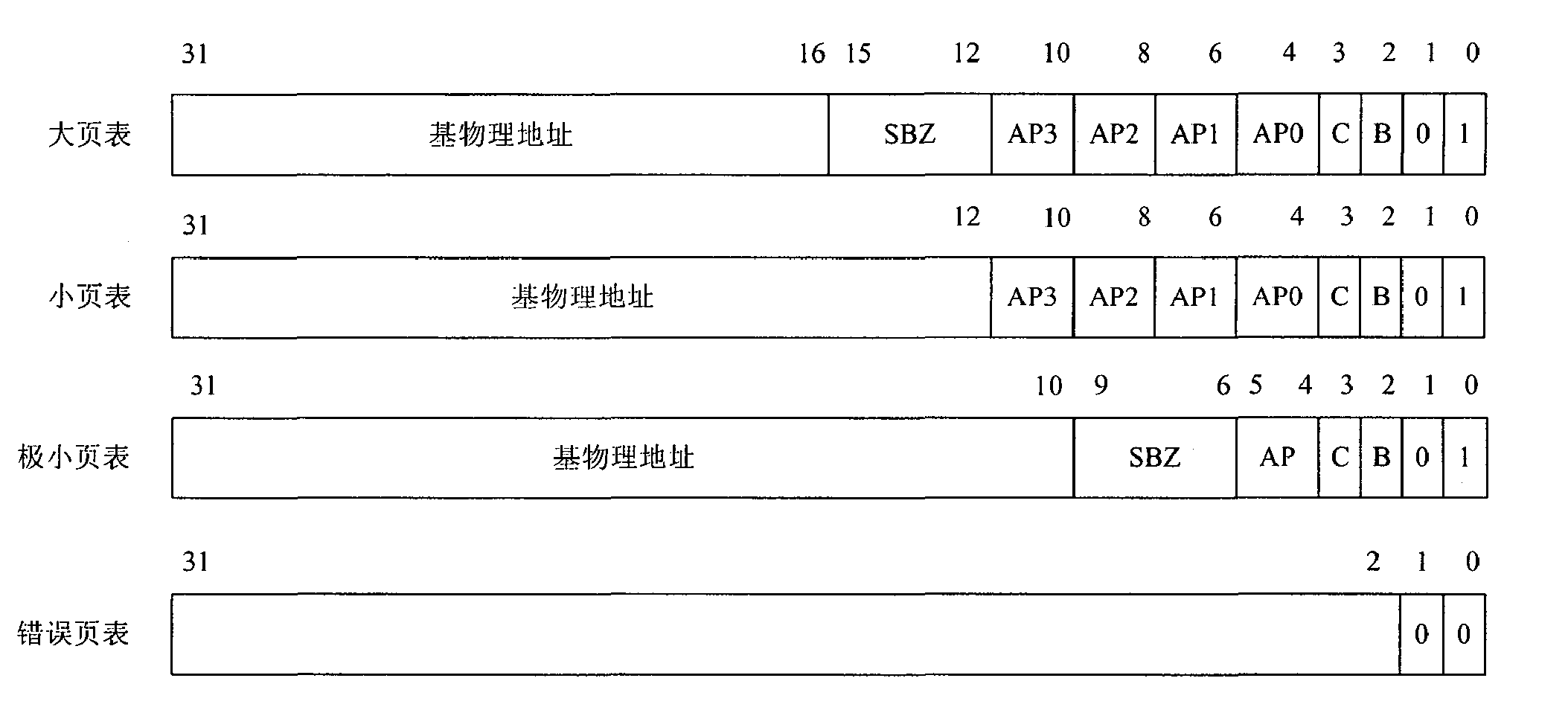


图2-2 L2页表项

三、多级页表如何进行地址转换？

1. 基于一级页表的的地址变换过程

CP15的寄存器C2中存放的是内存中页表的基地址。其中位[31:14]为内存中页表的页表的基地址，位[13:0]为 0。因此一级页表的基地址必须是16KB对齐的。CP15的寄存器C2的位[31:14]和虚拟地址的位[31:20]结合作为一个32位数的高30位，在将该32位数的低两位值为00，从而形成一个32位的索引值。使用该32位的索引值从页表中可以查到一个4字节的地址变换条目。该条目中或者包含了一个一级描述符，或者包含了一个指向二级页表的指针。

根据上面的过程可以得到页表中相应的地址变换条目。该条目称为一级描述符，一级描述符定义了与之对应的1M存储空间是如何映射的。一级描述符的位[1:0]定义了该一级描述符的类型，如果位[1:0]的值为二进制的10，该一级描述符为段描述符( section descriptor)，则此页表项含有一个有效的1MB页可用，页表项中的值与虚拟地址的偏移量部分合并来组成物理地址。如果[1:0]为00，相应的1MB虚拟存储空间没有被映射到物理存储空间，因而访问该存储空间将产生地址变换失效信号;如果[1:0]为01或者11,则MMU执行基于二级页表的地址变换过程。

(2)基于二级页表的地址变换过程

第一步，用L1偏移量部分索引L1主页表，找到虚拟地址的L1页表项。如果该页表项的最低2位是二进制的01，则表示该页表项包含的是一个粗页的L2页表基地址。

第二步，将L2偏移量部分合并到第一步找到的L2页表基地址，得到的地址用来选择包含所搜索页的转换数据的页表项。MMU将L2页表项的基地址与虛拟地址的偏移量部分合并起来组成所请求的物理存储器地址。

逻辑地址和物理地址的转换过程如图2-3所示：

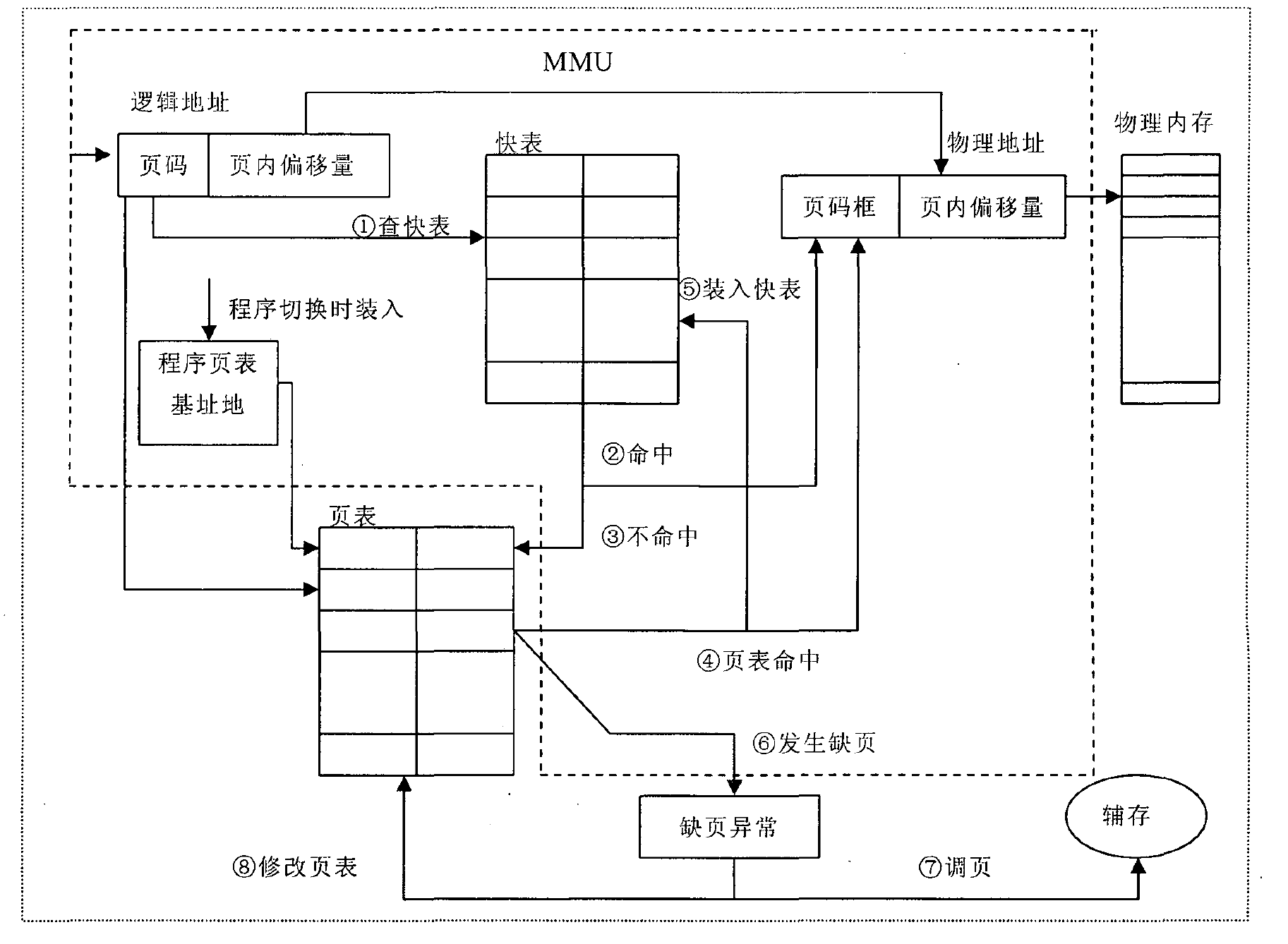


图2-3 逻辑地址和物理地址的转换过程图

四、设置TLB以及使用TLB前后的访问时间

1. ARM中的TLB

TLB：在ARM系统中，MMU中临时存储转换数据的一组重定位寄存器实际上是由一个64个重定位寄存器组成的全相联cache，即TLB。TLB缓冲最近被访问的页的转换数据。

添加TLB后，CPU访问内存的过程如图2-4.

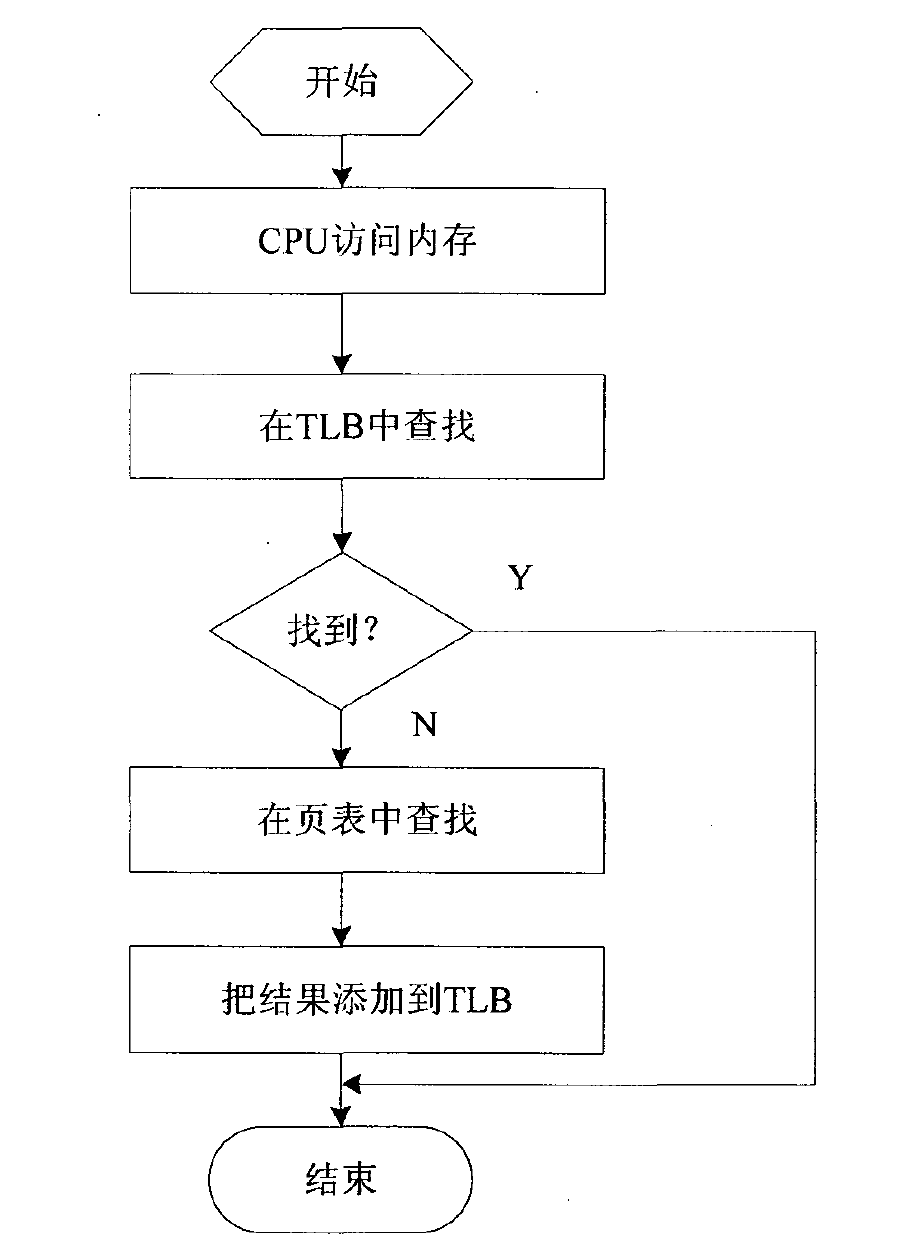


图2-4 CPU访问内存的过程

2. 添加TLB前后效率对比

λ：查找快表的时间

t：假设查页表找页表项和查地址找数据拼接地址的时间都分别为t

a: 快表命中率

由于ARM是两级页表结构，因此：

**i. 没有快表：**

EAT = t【查找一级页表】 + t【查找二级页表】+t【物理地址转换】 = 3t

**ii. 有快表：**

EAT = 【快表中找到】a\*λ +【快表没有找到】(1-a)\*(2t【在内存中找（页目录和页表）】+λ【更新快表】)+t【转换为物理地址】

五、参考文献

[1]王宏宇. ARM32位处理器存储器管理单元系统级仿真模型的研究[D].华北电力大学（北京）,2009.

[2]<https://blog.csdn.net/qq_34160841/article/details/105611131>

[3]https://developer.arm.com/documentation/den0024/a/The-Memory-Management-Unit/The-Translation-Lookaside-Buffer