ARM 中 MMU 工作原理

本文描述基于存储器管理单元的系统结构, 包含以下内容：

- 关于存储器管理单元的结构
- 存储器访问的顺序
- 转换过程
- 访问权限
- 域
- 异常
- CP15 寄存器

3.1 关于存储器管理单元的结构

MMU 存储器系统的结构允许对存储器系统的精细控制。大部分的控制细节由存在存储器中的转换表提供。这些表的入口定义了从 1KB 到 1MB 的各种存储器区域的属性。这些属性包括：

虚拟地址到物理地址映射

ARM 处理器产生的地址叫虚拟地址，MMU 允许把这个虚拟地址映射到一个不同的物理地址去。这个物理地址表示了被访问的主存储器的位置。

它允许用很多方式管理物理存储器的位置，例如：它可以用具有潜在冲突的地址映射为不同的进程分配存储器，或允许具有不连续地址的应用把它映射到连续的地址空间。

------注------

如果使用了快速上下文切换扩展(Fast Context Switch Extension)，则在

本文中的虚拟地址的意思应该是修改过的虚拟地址(Modified virtual

address)

存储器访问权限(permissions)

这些控制对存储器区域的不可访问权限、只读权限、读写权限。当访问不可

访问权限的存储器时，会有一个存储器异常通知 ARM 处理器。

允许权限的级别也受程序运行在用户状态还是特权状态影响，还受是否使用

了域有关。

高速缓存和缓冲位(Cachability and bufferability bits [C and B])

这些在高速缓存和缓冲一节讲

系统控制协处理器的寄存器允许对系统的高级控制,如转换表的位置。他们也用来

为 ARM 提供内存异常的状态信息。

查找整个转换表的过程叫转换表遍历。它由硬件制动进行，并需要大量的执行时间

（至少一个存储器访问，通常是两个)。为了减少存储器访问的平均消耗， 转换表

遍历结果被高速缓存在一个或多个叫作 Translation Lookaside Buffers(TLBs)的

结构中。通常在 ARM 的实现中每个内存接口有一个 TLB。

- 有一个存储器接口的系统通常有一个唯一的 TLB
- 指令和数据的内存接口分开的系统通常有分开的指令 TLB 和数据 TLB

如果系统有高速缓存， 高速缓存的数量也通常是由同样的方法确定的。所以在高

速缓存的系统中，每个高速缓存一个 TLB。

当存储器中的转换表被改变或选中了不同的转换表(通过写 CP15 的寄存器 2)，先前高速缓存的转换表遍历结果将不再有效。MMU 结构提供了刷新 TLB 的操作。

MMU 结构也允许特定的转换表遍历结果被锁定在一个 TLB 中，这就保证了对相关的存储器区域的访问绝不会导致转换表遍历，这也对那些把指令和数据锁定在高速缓存中的实时代码有相同的好处。

3.2 存储器访问的顺序

当 ARM 要访问存储器时，MMU 先查找 TLB 中的虚拟地址表，如果 ARM 的结构支持分开的地址 TLB 和指令 TLB，那么它用：

- 取指令使用指令 TLB
- 其它的所有访问类别用数据 TLB

如果 TLB 中没有虚拟地址的入口，则转换表遍历硬件从存在主存储器中的转换表中获取转换和访问权限，一旦取到，这些信息将被放在 TLB 中，它会放在一个没有使用的入口处或覆盖一个已有的入口。关于转换表的信息和转换表遍历的实现参见转换过程一节。

一旦为存储器访问的 TLB 的入口被拿到,这些信息将被用于：

1. C（高速缓存）和 B（缓冲）位被用来控制高速缓存和写缓冲，并决定是否高速缓存。（如果系统中没有高速缓存和写缓冲，则对应的位将被忽略）

2. 访问权限和域位用来控制访问是否被允许。如果不允许，则 MMU 将向 ARM 处理器发送一个存储器异常；否则访问将被允许进行。

访问权限、域和异常几节有详细描述。

3. 对没有高速缓存的系统（包括在没有高速缓存系统中的所有存储器访问），物理地址将被用作主存储器访问的地址。

对有高速缓存的系统，在高速缓存没有选中的情况下，物理地址将被用行取(line fetch)的地址。如果选中了高速缓存，则物理地址将被忽略。

图 3-1 说明了这种高速缓存系统

访问控

制硬件

TLB

ARM

处理器

高速缓

存和写

缓冲

转换表遍历

硬件

高速缓

存行取

硬件

主

存

储

器

虚拟地址

异常

域位

C, B 位

物理地址

图 3-1 高速缓存的 MMU 存储器系统

3.2.1 允许和禁止 MMU

通过写系统控制协处理器的寄存器 1 的第 0 位可以允许和禁止 MMU。在复位后这位是 0，MMU 被禁止。

当 MMU 被禁止时，存储器访问将被按如下处理：

1. 由具体的实现确定当 MMU 被禁止时是否能够允许高速缓存和写缓冲。

- 当 MMU 被禁止时不能允许高速缓存和写缓冲时，C 和 B 位不起作用。
- 当 MMU 被禁止时能允许高速缓存和写缓冲时：

i. 访问数据时被认为没有高速缓存和写缓冲（C==0，B==0）

ii. 取指令时：

a) 当系统只有一个唯一的 TLB 时，认为是没有高速缓存。（C==0）

b) 当系统只有独立的指令 TLB 时，认为是有高速缓存。（C==1）

2. 没有存储器访问权限的检查，MMU 也不产生异常信号。

3. 物理地址与虚拟地址相同（即所谓的平坦地址映射模式）。

在允许 MMU 之前，必须在内存中建立适当的转换表，并且所有相关的 CP15 寄存器要被初始化正确。

注：----------

允许和禁止 MMU 直接改变了虚拟地址到物理地址的映射（除非转换表被设定为平坦地址映射模式）。所以很可能在允许 MMU 时所有的高速缓存需要被刷新。

另外，如果允许 MMU 的指令的物理地址和虚拟地址不同，取指令将变得复杂化。所以，强烈建议允许 MMU 的指令具有相同的物理地址和虚拟地址。

3.3 转换过程

MMU 支持基于节或页的存储器访问：

节（Section） 构成 1MB 的存储器块

支持 3 中不同的页尺寸：

微页（Tiny page） 构成 1KB 的存储器块

小页（Small page） 构成 4KB 的存储器块

大页（Large page） 构成 64KB 的存储器块

节和大页是支持允许只用一个 TLB 入口去映射大的存储器区间。小页和大页有附加的访问控制：小页分成 1KB 的子页，和大页分成 16KB 的子页。微页没有子页，对微页的访问控制是对整个页。

存在主存储器内的转换表有两个级别：

第一级表 存储节转换表和指向第二级表的指针。

第二级表 存储大页和小页的转换表。一种类型的第二级表存储微页转换表。

MMU 把 CPU 产生的虚拟地址转换成物理地址去访问外部存储器，同时继承并检查访问权限。地址转换有四条路径。路径的选取由这个地址是被标记成节映射访问还是页映射访问确定。页映射访问可以是大、小和微页的访问。

然而，转换过程总是由下面所描述的那样由第一级表的获取开始。节映射的访问只需要读取第一级表，页映射的访问还需要读取第二级表。

3.3.1 转换表基址

当片上（on-chip）的 TLB 中不包含被要求的虚拟地址的入口时，转换过程被启动。转换表基址寄存器（CP15 的寄存器 2）保存着第一级转换表基址的物理地址。只有 bits[31:14]有效，bits[13:0]应该是零（SBZ）。所以第一级表必须在 16KB 的边界。

3.3.2 取第一级表

转换表基址寄存器的 bits[31:14]与虚拟地址的 bits[31:20]和两个 0 位连接形成 32 为物理地址，如图 3-2。这个地址选择了一个四字节的转换表入口，它是第一级描述符或是指向第二级页表的指针。

转换基址 SBZ

转换基址 表索引 00

31 14 13 0

31 20 19 0

31 14 13 2 10

表索引 xxxxxxxxxxxxxxxxxxxxxx

图 3-2 访问转换表的第一级描述符

3.3.3 第一级描述符

第一级表的每个入口是一个描述它所关联的 1MB 虚拟地址是如何映射的描述符。见表 3-1，根据 bits[1:0]的组合，有四种可能：

- 如果 bits[1:0]==0b00，所关联的地址没有被映射，试图访问他们将产生一个转换错（fault）。因为他们被硬件忽略，所以软件可以利用这样的描述符的 bits[31:2]做自己的用途。推荐为描述符继续保持正确的访问权限。
- 如果 bits[1:0]==0b10，这个入口是它所关联地址的节描述符。见节描述符和转换节参考中的细节。
- 如果 bits[0]==1，这个入口给出粗糙第二级表（bit[1]==0），或精细第二级表（bit[1]==1）。每一种类型的表描述了它所关联的 1MB 存储区域的映射。粗糙第二级表较小，每个表 1KB，每个精细第二级表 4KB。然而粗糙第二级表只能映射大页和小页，精细第二级表可以映射大页、小页和微页。

3.3.4 节描述符和转换节参考

如果第一级描述符是节描述符，那么各个字段有如下的意义：

Bits[1:0] 描述符类型标识（0b10 表示节描述符）

Bits[3:2] 高速缓存和缓冲位

Bits[4] 由具体实现定义

Bits[8:5] 这个描述符控制的节的 16 种域之一

Bits[9] 现在没有使用，应该为零

Bits[11:10] 访问控制，见表 3-3

Bits[19:12] 现在没有使用，应该为零

Bits[31:20] 节基址，形成物理地址的高 12 位

忽略 00

粗糙页表基址 sbz 域 imp 00

节基址 SBZ AP sbz 域 imp C B 10

精细页表基址 SBZ 域 imp 11

31 20 19 12 11 10 9 8 5 4 3 2 10

表 3-1 第一级描述符格式

错

粗糙页表

节

精细页表

图 3-3 表示了节转换的完整过程。

注：----------

访问权限必须在物理地址产生之前去检查，检查访问权限的顺序见访问权限一节。

表索引 节索引

31 20 19 0

虚拟地址

转换基址 SBZ

31 14 13 0

转换表基址

转换基址 表索引 00

31 14 13 2 1 0

第一级表地址

节基址 SBZ AP sbz 域 imp C B 10

31 20 19 12 11 10 9 8 5 4 3 2 1 0

第一级表描述符

节基址 节索引

31 20 19 0

物理地址

图 3-3 节转换

First-level fetch

3.3.5 粗糙页表描述符

如果第一级描述符是粗糙页表描述符，那么各个字段有如下的意义：

Bits[1:0] 描述符类型标识（0b01 表示粗糙页表描述符）

Bits[4:2] 由具体实现定义

Bits[8:5] 这个描述符控制的页的 16 种域之一

Bits[9] 现在没有使用，应该为零

Bits[31:10] 页表基地址是一个指向第二极粗糙页表的指针，它给出第二级表访问的基地址。而第二级粗糙页表必须在 1KB 边界对齐。

如果从第一级读取到的是二级粗糙页表描述符，那么会象图 3-4 所示执行第二级描述符读取。

第一级表索引 第二级表索引 xxxxx

31 20 19 12 11 0

虚拟地址

转换基址 SBZ

31 14 13 0

转换表基址

转换基址 第一级表索引 0 0

31 14 13 2 1 0

第一级描述符地址

页表基址 sbz 域 imp 01

31 10 9 8 5 4 2 1 0

第一级描述符

页表基址 第二级表索引 00

31 10 9 2 1 0

第二级描述符地址

First-level fetch

图 3-4 访问粗糙页表第二级描述符

3.3.6 精细页表描述符

如果第一级描述符是精细页表描述符，那么各个字段有如下的意义：

Bits[1:0] 描述符类型标识（0b11 表示精细页表描述符）

Bits[4:2] 由具体实现定义

Bits[8:5] 这个描述符控制的页的 16 种域之一

Bits[11:9] 现在没有使用，应该为零

Bits[31:10] 页表基地址是一个指向第二级精细页表的指针，它给出第二级表访问的基地址。而第二级精细页表必须在 4KB 边界对齐。

如果从第一级读取到的是二级精细页表描述符，那么会象图 3-5 所示执行第二级描述符读取。

第一级表索引 第二级表索引 xxxxx

31 20 19 10 9 0

虚拟地址

转换基址 SBZ

31 14 13 0

转换表基址

转换基址 第一级表索引 0 0

31 14 13 2 1 0

第一级描述符地址

页表基址 sbz 域 imp 01

31 12 11 9 8 5 4 2 1 0

第一级描述符

页表基址 第二级表索引 00

31 12 11 2 1 0

第二级描述符地址

First-level fetch

图 3-5 访问精细页表第二级描述符

3.3.7 第二级描述符

每个粗糙第二级表对映着以 4KB 为单位的虚拟地址范围市怎么映射的，每个精细第二级表对映着以 1KB 为单位的虚拟地址范围市怎么映射的。那些入口是页描述符，他们能够分别描述大于 4KB 或 1KB 的页。在这种情况下，这个描述符必须被重复足够次，以保证这个页始终使用相同的描述符，不论访问这个页中的哪个虚拟地址。

对于一个第二级描述符，有四种可能，由描述符的 bits[1:0]选择。见表 3-2：

- 如果 bits[1:0]==0b00，说关联的虚拟地址没有被映射，任何对这些虚拟地址的访问将会导致转换错(fault)。软件可以利用这样的描述符的 bits[31:2]做自己的用途，因为他们被硬件忽略。推荐为描述符继续保持正确的访问权限。
- 如果 bits[1:0]==0b01，这个入口是大页描述符，描述 64KB 的虚拟地址。见转换大页参考。

一个大页描述符在精细第二级表中必须被重复 64 次，在粗糙第二级表中必

须被重复 16 次以保证所有的虚拟地址都被描述。

• 如果 bits[1:0]== 0b10，这个入口是小页描述符，描述 4KB 的虚拟地址。

见转换小页参考。

一个小页描述符在精细第二级表中必须被重复 4 次，以保证所有的虚拟地址都被描述。在粗糙第二级表中只有一个实例。

• 如果 bits[1:0]== 0b11，这个入口是微页描述符，描述 1KB 的虚拟地址。

见转换微页参考。

在精细第二级表中只需要一个微页描述符的实例。微页描述符不能在粗糙第二级表中出现，如果出现了，结果不可预测。

忽略 00

大页基地址 SBZ AP3 AP2 AP1 AP0 C B 01

小页基地址 AP3 AP2 AP1 AP0 C B 01

微页基地址 SBZ AP C B 11

表 3-2 第二级描述符格式

31 16 15 12 11 10 9 8 7 6 5 4 3 2 1 0

错

大页

小页

微页

大页描述符字段

大页描述符的字段有如下意义：

bits[1:0] 表示描述符的类型

bits[3:2] 高速缓促和缓冲位

bits[11:4] 访问权限位。这些为控制对页的访问。关于这些位的解释见表 3-3。

大页被分成 4 各子页。

AP0 编码对第一个子页的访问权限。

AP1 编码对第二个子页的访问权限。

AP2 编码对第三个子页的访问权限。

AP3 编码对第四个子页的访问权限。

bits[15:12] 现在没有使用，应该为零。

bits[31:16] 用来形成物理地址的对应位。

小页描述符字段

小页描述符的字段有如下意义：

bits[1:0] 表示描述符的类型

bits[3:2] 高速缓促和缓冲位

bits[11:4] 访问权限位。这些为控制对页的访问。关于这些位的解释见表 3-3。

小页被分成 4 各子页。

AP0 编码对第一个子页的访问权限。

AP1 编码对第二个子页的访问权限。

AP2 编码对第三个子页的访问权限。

AP3 编码对第四个子页的访问权限。

bits[31:12] 用来形成物理地址的对应位。

微页描述符字段

微页描述符的字段有如下意义：

bits[1:0] 表示描述符的类型

bits[3:2] 高速缓促和缓冲位

bits[5:4] 访问权限位。这些为控制对页的访问。关于这些位的解释见表 3-3 关

于微页的解释。

bits[9:6] 现在没有使用，应该为零。

bits[31:10] 用来形成物理地址的对应位。

3.3.8 转换大页参考

图 3-6 显示了在粗糙第二级表中转换一个 64KB 的大页的完整顺序。在精细第二级表中的转换顺序页相似，只是第二级描述符的地址如精细页表描述符一节所决定。

注：----------

页索引的高 4 位和第二级表的低阶 4 位重叠，在粗糙页表中大页的每个页表入口必须被重复 16 次。在精细页表中大页的每个页表入口必须被重复 64 次。

大页基址 页索引

31 16 15 0

物理地址

大页基址 SBZ AP3 AP2 AP1 AP0 C B 0 1

31 16 15 121110 9 8 7 6 5 4 3 2 10

0

第二级描述符

页表基址 第二级表索引 00

31 10 9 2 10

第二级描述符地址

页表基址 sbz 域 imp 01

31 10 9 8 5 4 2 10

第一级描述符

转换基址 第一级表索引 00

31 14 13 2 10

第一级描述符地址

转换基址 SBZ

31 14 13 2 10

转换表基址

第一级表索引 第二级表索引 页索引

31 20 19 16 15 12 11 0

虚拟地址

Second-level fetch

First-level fetch

图 3-6 粗糙第二级表中的大页转换

3.3.9 转换小页参考

图 3-7 显示了在粗糙第二级表中转换一个 4KB 的小页的完整顺序。在精细第二级表中的转换顺序页相似，只是第二级描述符的地址如精细页表描述符一节所决定。

注：----------

当小页出现在精细第二级表中时，页索引的高 2 位和第二级表的低阶 2 位重叠，在精细页表中小页的每个页表入口必须被重复 4 次。

图 3-7 粗糙第二级表中的小页转换

小页基址 页索引

31 12 11 0

物理地址

小页基址 AP3 AP2 AP1 AP0 C B 10

31 12 11 10 9 8 7 6 5 4 3 2 10

0

第二级描述符

页表基址 第二级表索引 00

31 10 9 2 10

第二级描述符地址

页表基址 sbz 域 imp 01

31 10 9 8 5 4 2 10

第一级描述符

转换基址 第一级表索引 00

31 14 13 2 10

第一级描述符地址

第一级表索引 第二级表索引 页索引

31 20 19 12 11 0

虚拟地址

转换基址 SBZ

31 14 13 0

转换表基址

First-level fetch

Second-level fetch

3.3.10 转换微页索引

图 3-8 显示了在精细第二级表中转换 1KB 微页的完整过程。

注：----------

微页不能出现在粗糙第二级表中。

转换基址 SBZ

31 14 13 0

转换表基址

第一级表索引 第二级表索引 页索引

31 20 19 10 9 0

虚拟地址

转换基址 第一级表索引 00

31 14 13 2 10

第一级描述符地址

页表基址 sbz 域 imp11

31 12 11 9 8 5 4 2 10

第一级描述符

微页表基址 SBZ AP C B 11

31 10 9 6 5 4 3 2 10

第二级描述符

页表基址 第二级表索引 00

31 12 11 2 10

第二级描述符地址

微页表基址 页索引

31 10 9 0

物理地址

Second-level fetch

First-level fetch

图 3-8 精细第二级表中的微页转换

3.4 访问权限

在节和页描述符中的访问权限位控制对相应的节和页的访问。访问权限由 CP15 的寄存器 1 的 System(S)和 ROM(R)位修改。表 3-3 描述了访问权限位和 S、R 位相互作用时的意义。如果访问了没有访问权限的存储器空间，将会产生权限错（见异常一节）。

表 3-3 MMU 访问权限

AP

S

R

Privileged permissions

User permissions

0b00 0 0 不能访问 不能访问

0b00 1 0 只读 不能访问

0b00 0 1 只读 只读

0b00 1 1 不可预测 不可预测

0b01 X X 读 / 写 不能访问

0b10 X X 读 / 写 只读

0b11 X X 读 / 写 读 / 写

3.5 域

域是节、大页和小页的集合。ARM 结构支持 16 个域。对域的访问由域访问控制寄存器的两个位字段控制。因为每个字段对访问对应的域的使能非常迅速，所以整个

存储器区间能很快地交换进出虚拟存储器。这里支持 2 种域访问方式：

客户 域的用户（执行程序，访问数据），被形成这个域的节或页来监督访

问权限。

管理者 控制域的行为（域中的当前节和页，对域的访问），不被形成这个域

的节或页来监督访问权限。

一个程序可以是一些域的客户，也是另外一些域的管理者，同时没有对其它域的访问权限。这允许对程序访问不同存储器资源的非常灵活的存储器保护。表 3-4 说明了域访问控制寄存器的位编码方式。

表 3-4 域访问的值

值

访问方式

描述

0b00 不能访问 任何访问都将导致一个域错(domain fault)

0b01 客户 能否访问将根据节或页描述符中的访问权限位

确定

0b10 保留 使用这个值将导致不可预料的结果

0b11 管理者 不根据节或页描述符中的访问权限位确定能否

访问，所以不会产生权限错。(permission

fault)