(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 110688330 B
(45)授权公告日 2021.08.31

(21)申请号 201910898772.4

(22)申请日 2019.09.23

(65)同一申请的已公布的文献号
申请公布号 CN 110688330 A

(43)申请公布日 2020.01.14

(73)专利权人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 白跃彬 禹超

(51)Int.Cl.
G06F 12/1027(2016.01)
G06F 12/0882(2016.01)

(56)对比文件
CN 109933441 A,2019.06.25
CN 103562854 A,2014.02.05
US 2014149652 A1,2014.05.29

杨婷.高性能嵌入式CPU旁路转换单元设计.《中国优秀硕士学位论文全文数据库 信息科技辑》.2014,(第07期),

W. Jiang.Beyond TCAMs: An SRAM-Based Parallel Multi-Pipeline Architecture for Terabit IP Lookup.《IEEE INFOCOM 2008 - The 27th Conference on Computer Communications》.2008,

审查员 邱祥吉

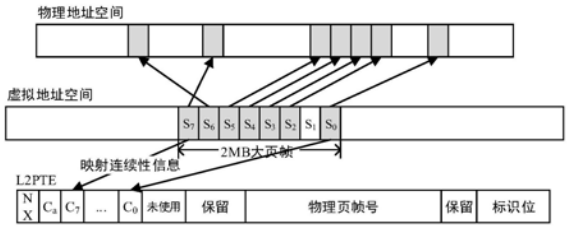
权利要求书3页 说明书5页 附图3页

(54)发明名称

一种基于内存映射相邻性的虚拟内存地址翻译方法

(57)摘要

本发明公开了一种基于内存映射相邻性的虚拟内存地址翻译方法,包括:1)将每个2MB的虚拟地址空间分为8个等长的内存子域;2)操作系统分配内存空间时,扫描页表,识别出连续映射的内存子域,并将内存子域内部和内存子域间的映射连续性信息保存在2级页表项中的未使用位;3)在地址翻译过程中查找页表时,依据保存在2级页表项中的映射连续性信息合并连续映射内存域的地址翻译信息并保存在TLB中;4)为进一步提升可合并地址翻译信息的页面数量,在查找页表时同时检测连续映射的内存子域间的连续性,并将内存子域间连续性信息保存在内存子域高速缓存中以减少内存访问次数。本发明能够有效提高TLB的覆盖范围、提升TLB的命中率,从而降低虚拟内存地址翻译的开销。



CN 110688330 B

1.一种基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于,包括以下步骤:

(1)将进程虚拟地址空间中每个2MB大页帧的地址空间划分为8个等长的子地址空间,其中的每个子地址空间称为内存子域,在操作系统为进程分配物理内存空间后,扫描进程页表并识别出连续映射的内存子域,最后,将内存子域的映射连续性信息保存在扩展后的相应2级页表项L2PTE中;

(2)当处理器执行访存指令时,需首先执行虚拟地址到物理地址的翻译操作,当所请求的地址翻译信息不存在于旁路转换缓存TLB中时,就会执行页表查询操作;

(3)在页表查询过程中,根据L2PTE中的映射连续性信息判断请求地址所属虚拟页VPN_{req}是否位于连续映射内存子域中,并根据连续性信息进一步得到VPN_{req}所映射的物理页号PPN_{req};

(4)得到请求地址所属虚拟页号VPN_{req}映射的物理页号PPN_{req}后,依据页内偏移地址得到所请求虚拟地址映射的物理地址,并返回执行单元;

(5)此外,如果请求地址所属虚拟页VPN_{req}位于连续映射内存子域中,则将合并后的连续映射内存域的地址翻译信息保存在TLB中;否则,只需将请求地址所属虚拟页号VPN_{req}到物理页号PPN_{req}的映射关系保存在TLB中;

(6)当需要执行地址翻译操作并查询TLB时,会依据TLB项的类型判断所请求地址是否在TLB项中地址翻译信息的覆盖范围,如存在TLB项包含了所请求地址的翻译信息,则可依据TLB项中的地址翻译信息得到真实的物理地址,否则,执行步骤(2)。

2.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于:所述步骤(1)中,扫描进程页表并识别出连续映射的内存子域,其中判断内存子域是否连续映射的条件是内存子域所代表的一段连续虚拟地址空间是否被映射到一段连续物理地址空间;其中内存子域的映射连续性信息保存在扩展后的相应L2PTE中的方法为:

(1)对进程页表的每个L2PTE,依次判断其所指向的2MB虚拟大页帧中的8个内存子域的映射连续性,以及整个2MB虚拟大页帧的映射连续性;

(2)如果一个内存子域是连续映射的,则将其所属的L2PTE中对应的内存子域状态位C_n置1;

(3)如果整个2MB虚拟大页帧被映射到连续的2MB物理地址空间,则将其L2PTE中的C_a置1。

3.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于:所述步骤(1)中,将内存子域的映射连续性信息保存在扩展后的相应L2PTE位中,L2PTE的扩展方式为:

(1)在L2PTE高位未被利用的位中使用8位C_0～C_7来分别保存对应的8个内存子域的连续性信息;

(2)在L2PTE高位未被利用的位中使用1位C_a用来保存对应2MB虚拟大页帧的映射连续性信息。

4.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于,所述步骤(3)中,根据L2PTE中的映射连续性信息判断请求地址所属虚拟页VPN_{req}是否位于连续映射内存子域中,其具体判断方法为:

(1)在读取L2PTE时,如果C_a位为1,则表示所请求虚拟页VPN_{req}不断属于连续映射内存子

域中，而且属于连续映射的2MB虚拟大页帧；

(2)如果C_a位为0，但是虚拟页VPN_{req}所在的内存子域对应的C_i位为1，则表示所请求虚拟页VPN_{req}属于连续映射内存子域；

(3)如果C_a位为0，而且虚拟页VPN_{req}所在的内存子域对应的C_i位也为0，则表示所请求虚拟页VPN_{req}不属于连续映射内存子域。

5.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法，其特征在于，所述步骤(3)中，根据连续性信息进一步得到VPN_{req}所映射的物理页号PPN_{req}，其具体方法为：

(1)当请求虚拟页VPN_{req}属于连续映射的2MB虚拟大页帧时，只需要读取其L2PTE所指向的第一个1级页表项，在获得第一个1级页表项中保存的物理页号PPN_0后，根据第一个1级页表项所指向的虚拟页号VPN_0就可以得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}：$PPN_{req}=PPN_0+(VPN_{req}-VPN_0)$；

(2)当请求虚拟页VPN_{req}不属于连续映射的2MB虚拟大页帧，但属于连续映射内存子域S_{cur}时，则依据当前内存子域S_{cur}与其所有邻近内存子域间的连续性得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}；

(3)当请求虚拟页VPN_{req}不属于连续映射内存子域时，直接读取其对应的1级页表项，其中保存的物理页号就是虚拟页VPN_{req}映射的物理页号PPN_{req}。

6.根据权利要求5所述的基于内存映射相邻性的虚拟内存地址翻译方法，其特征在于，所述步骤(2)中，依据当前内存子域S_{cur}与其所有邻近内存子域间的连续性得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}，其具体步骤为：

(1)读取当前2MB虚拟大页帧中所有与当前内存子域S_{cur}邻近的连续映射内存子域的第一个虚拟页的1级页表项，直到遇到非连续映射内存子域为止；

(2)根据这些连续内存子域的第一个虚拟页的1级页表项中的物理页号判断哪些邻近的连续映射内存子域与当前内存子域S_{cur}能够组成较大连续连续映射内存域，然后依据所组成的较大连续映射内存域中的第一个虚拟页号VPN_{s0}及其映射的物理页号PPN_{s0}就可得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}：$PPN_{req}=PPN_{s0}+(VPN_{req}-VPN_{s0})$；

为减少步骤(1)中读取1级页表项的访存次数，将内存子域间的连续性信息保存在内存子域高速缓存MSC中，其中MSC项包含如下字段：

(1)Tag：标签字段，保存虚拟大页帧号；

(2)Contiguity_Info：相邻性信息字段，长度为7比特位，其中每位表示相应的两个相邻内存子域间是否连续，某一位置1的条件是：相应的两个相邻内存子域都是连续映射的，而且这两个相邻的内存子域可以组成一个更大的连续映射内存域；

(3)Flags：标识字段，保存有效性信息和置换算法相关数据。

7.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法，其特征在于，所述步骤(5)中，将合并后的连续映射内存域的地址翻译信息保存在TLB中，其具体合并方法为：

(1)当请求虚拟页VPN_{req}属于连续映射的2MB虚拟大页帧时，将该2MB地址空间的第一个内存子域全局编号、该2MB虚拟大页帧中内存子域的数量以及该2MB虚拟大页帧的第一个虚拟页映射的物理页号保存在TLB项中；

(2)当请求虚拟页VPN_{req}不属于连续映射的2MB虚拟大页帧,但属于连续映射内存子域S_{cur}时,将当前内存子域S_{cur}与其邻近多个连续映射内存子域间组成的较大连续映射内存域中的第一个内存子域全局编号、该较大连续映射内存域中内存子域的数量以及该较大连续映射内存域中的第一个虚拟页映射的物理页号保存在TLB项中;

(3)当请求虚拟页VPN_{req}不属于连续映射内存子域中时,直接将请求地址的虚拟页VPN_{req}及其映射的物理页号PPN_{req}保存在TLB项中。

8.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于,所述步骤(6)中,依据TLB项的类型判断所请求地址是否在TLB项中地址翻译信息的覆盖范围,其中TLB项类型包括2种:(1)保存合并后地址翻译信息的聚合TLB项以及(2)保存常规虚拟页号到物理页号映射的常规TLB项;其中常规TLB项的格式不变,聚合TLB项包含如下字段:

(1)Tag:标签字段,保存内存子域全局编号;

(2)Length:长度字段,表示该TLB项聚合的连续映射内存子域的数量;

(3)Base:起始物理页号字段,表示该TLB项聚合的连续映射内存子域所映射的物理地址空间的初始物理页号;

(4)Flags:标识字段,保存有效性信息等。

9.根据权利要求1所述的基于内存映射相邻性的虚拟内存地址翻译方法,其特征在于,所述步骤(6)中,执行地址翻译操作并查询TLB,其中在查询TLB中的聚合TLB项时,如果该TLB项有效,其步骤如下:

(1)依据聚合TLB项中的Tag和Length字段计算出其所覆盖的虚拟页面的范围的最小虚拟页VPN_{min}以及最大虚拟页VPN_{max}:

VPN_{min}=Tag<<6

VPN_{max}=((Tag+Length)<<6)|0x3F

(2)如果请求虚拟页$VPN_{req}\in[VPN_{min},VPN_{max}]$,则说明请求虚拟页$VPN_{req}$所映射的物理页号$PPN_{req}$可以从该TLB项中得到:$PPN_{req}=Base+(VPN_{req}-VPN_{min})$;

(3)如果所有的TLB项中都不包含请求虚拟页VPN_{req}的映射信息,则需要进一步执行页表查询操作。

一种基于内存映射相邻性的虚拟内存地址翻译方法

技术领域

[0001] 本发明涉及虚拟内存管理和地址翻译等领域,特别是涉及一种基于内存映射相邻性的虚拟内存地址翻译方法。

背景技术

[0002] 虚拟内存技术能够有效提高进程的安全性以及可用内存容量,在基于虚拟内存的内存访问之前,需要将虚拟地址翻译成物理地址,然后才能使用物理地址来访问内存数据。在拥有4级页表的结构中,由于页表存储在内存中,地址翻译过程需要访问内存多达4次,这一过程非常耗时。长期以来,地址翻译操作一直被认为是处理器的性能瓶颈之一。虽然旁路转换缓存(Translation Lookaside Buffer,以下简称TLB)能够在一定程度上避免页表查询操作导致的多次内存访问,但是随着进程内存占用越来越大以及不规则访存操作越来越频繁,TLB对地址翻译操作性能的提升越来越有限。

[0003] 为了进一步提升虚拟内存效率、减少地址翻译开销,学术界和产业界开展了大量的研究工作。已有的具有代表性的研究成果主要分为以下两个方面:

[0004] (1)基于巨页的内存管理方法

[0005] 基于巨页的内存管理方法以巨页(如2MB或1GB)为单位对内存进行管理,TLB存储的地址翻译信息以巨页为单位,从而增加了TLB中地址翻译信息的覆盖范围、提高了TLB的命中率、极大地较少了地址翻译的开销。然而,巨页的支持通常会引入额外的内存管理开销,如会导致频繁的内存压缩和迁移。

[0006] (2)基于基础页的地址翻译合并方法

[0007] 基于基础页的地址翻译合并方法以基础页(4KB)为单位对内存进行管理,不需要巨页的支持,为了提升地址翻译效率,通常会将连续映射的多个相邻的基础页的地址翻译信息合并保存于TLB中,从而提升TLB的命中率、减少地址翻译的开销。然而,当前基于基础页的地址翻译合并方法只能合并有限数量基础页的地址翻译信息,为了合并尽可能多的基础页的地址翻译信息,就需要改变操作系统的内存管理策略来分配分配尽可能大的连续映射内存域。

[0008] 综上可以看出,为提升地址翻译效率,现有方法要么需要巨页的支持,要么需要修改操作系统的内存管理机制。本发明提出一种基于内存映射相邻性的虚拟内存地址翻译方法,该方法基于基础页,能够在不修改操作系统内存管理策略的前提下合并多达512个基础页的地址翻译信息,极大提升地址翻译的效率。

发明内容

[0009] 本发明技术解决问题:克服现有技术的不足和缺陷,提供一种基于内存映射相邻性的虚拟内存地址翻译方法,该方法基于基础页,能够在不修改操作系统内存管理策略的前提下合并多达512的基础页的地址翻译信息。

[0010] 本发明的技术解决方案,一种基于内存映射相邻性的虚拟内存地址翻译方法,包

括如下步骤：

[0011] (1)将进程虚拟地址空间划分为内存子域，在操作系统为进程分配物理内存空间后，扫描进程页表并识别出连续映射的内存子域，最后，将内存子域的映射连续性信息保存在扩展后的相应2级页表项(Level 2Page Table Entry，以下简称L2PTE)中；

[0012] (2)当处理器执行访存指令时，需首先执行虚拟地址到物理地址的翻译操作，当所请求的地址翻译信息不存在于旁路转换缓存(Translation Lookaside Buffer，以下简称TLB)时，就会执行页表查询操作；

[0013] (3)在页表查询过程中，根据L2PTE中的映射连续性信息判断请求地址所属虚拟页VPN_{req}是否位于连续映射内存子域中，并根据连续性信息进一步得到VPN_{req}所映射的物理页号PPN_{req}；

[0014] (4)得到请求地址所属虚拟页号VPN_{req}映射的物理页号PPN_{req}后，依据页内偏移地址得到所请求虚拟地址映射的物理地址，并返回执行单元；

[0015] (5)此外，如果请求地址所属虚拟页VPN_{reg}位于连续映射内存子域中，则将合并后的连续映射内存域的地址翻译信息保存在TLB中；否则，只需将请求地址所属虚拟页号VPN_{req}到物理页号PPN_{req}的映射关系保存在TLB中；

[0016] (6)当需要执行地址翻译操作并查询TLB时，会依据TLB项的类型判断所请求地址是否在TLB项中地址翻译信息的覆盖范围，如存在TLB项包含了所请求地址的翻译信息，则可依据TLB项中的地址翻译信息得到真实的物理地址，否则，执行步骤(2)。

附图说明

[0017] 图1是本发明的原理图；

[0018] 图2是常规页表项格式和本发明提出的扩展的2级页表项格式对比图；

[0019] 图3是基于内存子域映射连续性信息的页表查找的3种场景和方法；

[0020] 图4是内存子域高速缓存字段格式以及映射相邻性信息生成原理图；

[0021] 图5是常规TLB字段格式和融合TLB字段格式对比图；

具体实施方式

[0022] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本发明，并不用于限定本发明。此外，下面所描述的本发明各个实施方式中所涉及到的技术特征只要彼此之间未构成冲突就可以相互组合。

[0023] 本发明的基本思路在于，如图1所示，将虚拟地址空间中每个2MB的大页帧等分为8个固定大小的内存子域，在操作系统分配内存空间后，扫描页表并检测内存子域的映射连续性，并将映射连续性信息保存在L2PTE中；在地址翻译过程中通过解析保存在L2PTE中的映射连续性信息，将相邻的多个连续映射的内存子域的地址翻译信息合并保存在TLB中，从而增加TLB的命中率、提升地址翻译的效率。

[0024] 本发明基于内存映射相邻性的虚拟内存地址翻译方法包括以下步骤：

[0025] 1.将进程虚拟地址空间中每个2MB大页帧的地址空间划分为8个等长的子地址空间，其中的每个子地址空间称为内存子域，在操作系统为进程分配物理内存空间后，扫描进

程页表并根据内存子域所代表的一段连续虚拟地址空间是否被映射到一段连续物理地址空间判断内存子域的映射连续性，最后，将内存子域的映射连续性信息保存在扩展后的相应L2PTE位中，如图2所示，L2PTE中的扩展位包括：

[0026] (1)在L2PTE高位未被利用的位中使用8位(即C_0～C_7)用来分别保存对应的8个内存子域的连续性信息；

[0027] (2)在L2PTE高位未被利用的位中使用1位(即C_a)用来保存对应2MB虚拟大页帧的映射连续性信息

[0028] 内存子域的映射连续性信息保存在扩展后的相应L2PTE中的方法为：

[0029] (1)对进程页表的每个L2PTE，依次判断其所指向的2MB虚拟大页帧中的8个内存子域的映射连续性，以及整个2MB虚拟大页帧的映射连续性；

[0030] (2)如果一个内存子域是连续映射的，则将其所属的L2PTE中对应的内存子域状态位C_n置1；

[0031] (3)如果整个2MB虚拟大页帧被映射到连续的2MB物理地址空间，则将其L2PTE中的C_a置1。

[0032] 2.当处理器执行访存指令时，需首先执行虚拟地址到物理地址的翻译操作，当所请求的地址翻译信息不存在于旁路转换缓存(Translation Lookaside Buffer，以下简称TLB)时，就会执行页表查询操作；

[0033] 3.在页表查询过程中，根据L2PTE中的映射连续性信息判断请求地址所属虚拟页VPN_{reg}是否位于连续映射内存子域中，如图3所示，其具体判断方法为：

[0034] (1)在读取L2PTE时，如果C_a位为1，则表示所请求虚拟页VPN_{reg}不断属于连续映射内存子域中，而且属于连续映射的2MB虚拟大页帧；

[0035] (2)如果C_a位为0，但是虚拟页VPN_{reg}所在的内存子域对应的C_i位为1，则表示所请求虚拟页VPN_{reg}属于连续映射内存子域；

[0036] (3)如果C_a位为0，而且虚拟页VPN_{reg}所在的内存子域对应的C_i位也为0，则表示所请求虚拟页VPN_{reg}不属于连续映射内存子域。

[0037] 然后，根据连续性信息进一步得到VPN_{req}所映射的物理页号PPN_{req}，其具体方法为：

[0038] (1)当请求虚拟页VPN_{reg}属于连续映射的2MB虚拟大页帧时，只需要读取其L2PTE所指向的第一个1级页表项，在获得第一个1级页表项中保存的物理页号PPN_0后，根据第一个1级页表项所指向的虚拟页号VPN_0就可以得到所请求虚拟页VPN_{reg}映射的物理页号PPN_{req}：$PPN_{req}=PPN_0+(VPN_{req}-VPN_0)$；

[0039] (2)当请求虚拟页VPN_{reg}不属于连续映射的2MB虚拟大页帧，但属于连续映射内存子域S_{cur}时，则依据当前内存子域S_{cur}与其所有邻近内存子域间的连续性得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}；

[0040] (3)当请求虚拟页VPN_{reg}不属于连续映射内存子域时，直接读取其对应的1级页表项，其中保存的物理页号就是虚拟页VPN_{req}映射的物理页号PPN_{req}。

[0041] 以上步骤(2)中依据当前内存子域S_{cur}与其所有邻近内存子域间的连续性得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}的具体步骤为：

[0042] (1)读取当前2MB虚拟大页帧中所有与当前内存子域S_{cur}邻近的连续映射内存子域的第一个虚拟页的1级页表项，直到遇到非连续映射内存子域为止；

[0043] (2)根据这些连续内存子域的第一个虚拟页的1级页表项中的物理页号判断哪些邻近的连续映射内存子域与当前内存子域S_{cur}能够组成较大连续连续映射内存域,然后依据所组成的较大连续映射内存域中的第一个虚拟页号VPN_{s0}及其映射的物理页号PPN_{s0}就可得到所请求虚拟页VPN_{req}映射的物理页号PPN_{req}:$PPN_{req}=PPN_{s0}+(VPN_{req}-VPN_{s0})$。

[0044] 此外,为了减少以上步骤(1)中读取1级页表项的访存次数,将内存子域间的连续性信息保存在内存子域高速缓存(Memory Subregion Cache,以下简称MSC)中,如图4所示,其中MSC项包含如下字段:

[0045] (1)Tag:标签字段,保存虚拟大页帧号;

[0046] (2)Contiguity_Info:映射相邻性信息字段,长度为7比特位,其中每位表示相应的两个相邻内存子域间是否连续,某一位置1的条件是:相应的两个相邻内存子域都是连续映射的,而且这两个相邻的内存子域可以组成一个更大的连续映射内存域;

[0047] (3)Flags:标识字段,保存有效性信息等。

[0048] 4.得到请求地址所属虚拟页号VPN_{req}映射的物理页号PPN_{req}后,依据页内偏移地址得到所请求虚拟地址映射的物理地址,并返回执行单元;

[0049] 5.此外,如果请求地址所属虚拟页VPN_{reg}位于连续映射内存子域中,则将合并后的连续映射内存域的地址翻译信息保存在TLB中;否则,只需将请求地址所属虚拟页号VPN_{req}到物理页号PPN_{req}的映射关系保存在TLB中;其中,地址翻译信息的合并方法为:

[0050] (1)当请求虚拟页VPN_{reg}属于连续映射的2MB虚拟大页帧时,将该2MB地址空间的第一个内存子域全局编号、该2MB虚拟大页帧中内存子域的数量以及该2MB虚拟大页帧的第一个虚拟页映射的物理页号保存在TLB项中;

[0051] (2)当请求虚拟页VPN_{reg}不属于连续映射的2MB虚拟大页帧,但属于连续映射内存子域S_{cur}时,将当前内存子域S_{cur}与其邻近多个连续映射内存子域间组成的较大连续映射内存域中的第一个内存子域全局编号、该较大连续映射内存域中内存子域的数量以及该较大连续映射内存域中的第一个虚拟页映射的物理页号保存在TLB项中;

[0052] (3)当请求虚拟页VPN_{reg}不属于连续映射内存子域中时,直接将请求地址的虚拟页VPN_{req}及其映射的物理页号PPN_{req}保存在TLB项中。

[0053] 6.当需要执行地址翻译操作并查询TLB时,会依据TLB项的类型判断所请求地址是否在TLB项中地址翻译信息的覆盖范围,如存在TLB项包含了所请求地址的翻译信息,则可依据TLB项中的地址翻译信息得到真实的物理地址,否则,执行步骤(2)。

[0054] 其中,如图5所示,TLB项类型包括2种:(1)保存合并后地址翻译信息的聚合TLB项;(2)保存常规虚拟页号到物理页号映射的常规TLB项;常规TLB项的格式不变,聚合TLB项包含以下字段:

[0055] (1)Tag:标签字段,保存内存子域全局编号;

[0056] (2)Length:长度字段,表示该TLB项聚合的连续映射内存子域的数量;

[0057] (3)Base:起始物理页号字段,表示该TLB项聚合的连续映射内存子域所映射的物理地址空间的初始物理页号;

[0058] (4)Flags:标识字段,保存有效性信息等。

[0059] 在执行地址翻译操作并查询聚合TLB项时,如果该TLB项有效,则执行以下步骤来计算请求虚拟页VPN_{req}所映射的物理页号PPN_{req}:

[0060] (1)依据聚合TLB项中的Tag和Length字段计算出其所覆盖的虚拟页面的范围(最小虚拟页VPN_{min}以及最大虚拟页VPN_{max})：

[0061] VPN_{min}＝Tag＜＜6

[0062] VPN_{max}＝((Tag+Length)＜＜6)|0x3F

[0063] (2)如果请求虚拟页VPN_{req}∈[VPN_{min},VPN_{max}],则说明请求虚拟页VPN_{req}所映射的物理页号PPN_{req}可以从该TLB项中得到:PPN_{req}＝Base+(VPN_{req}-VPN_{min})；

[0064] (3)如果所有的TLB项中都不包含请求虚拟页VPN_{reg}的映射信息,则需要进一步执行页表查询操作。

[0065] 本发明未详细阐述部分属于本领域公知技术。

[0066] 以上所述,仅为本发明部分具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本领域的人员在本发明揭露的技术范围内,可轻易想到的变化或替换,都应涵盖在本发明的保护范围之内。

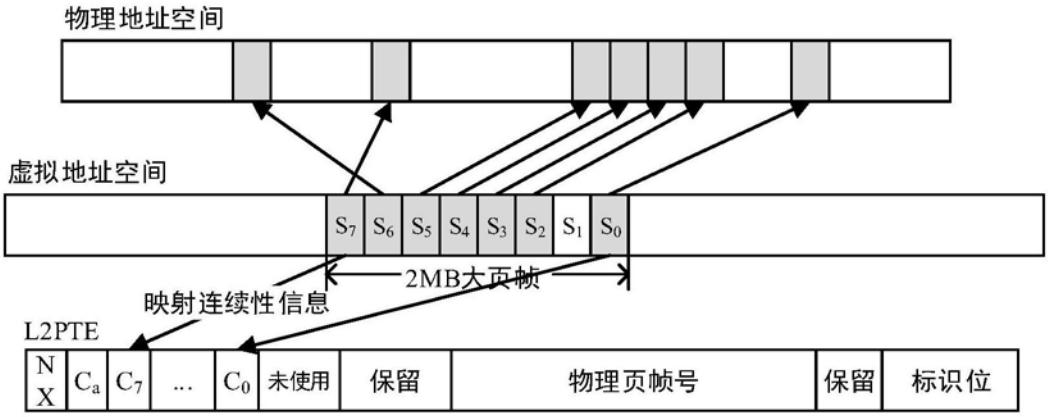

物理地址空间
虚拟地址空间
S7 S6 S5 S4 S3 S2 S1 S0
2MB大页帧
映射连续性信息
L2PTE
NX
Ca
C7
...
C0
未使用
保留
物理页帧号
保留
标识位

图1


63 62 51 45 11 8 0
NX
未使用
保留
物理页帧号
保留
标识位
(a) 常规PTE格式
63 62 61 60 54 53 51 45 11 8 0
NX
Ca
C7
...
C0
未使用
保留
物理页帧号
保留
标识位
(b) 扩展 L2PTE格式

图2

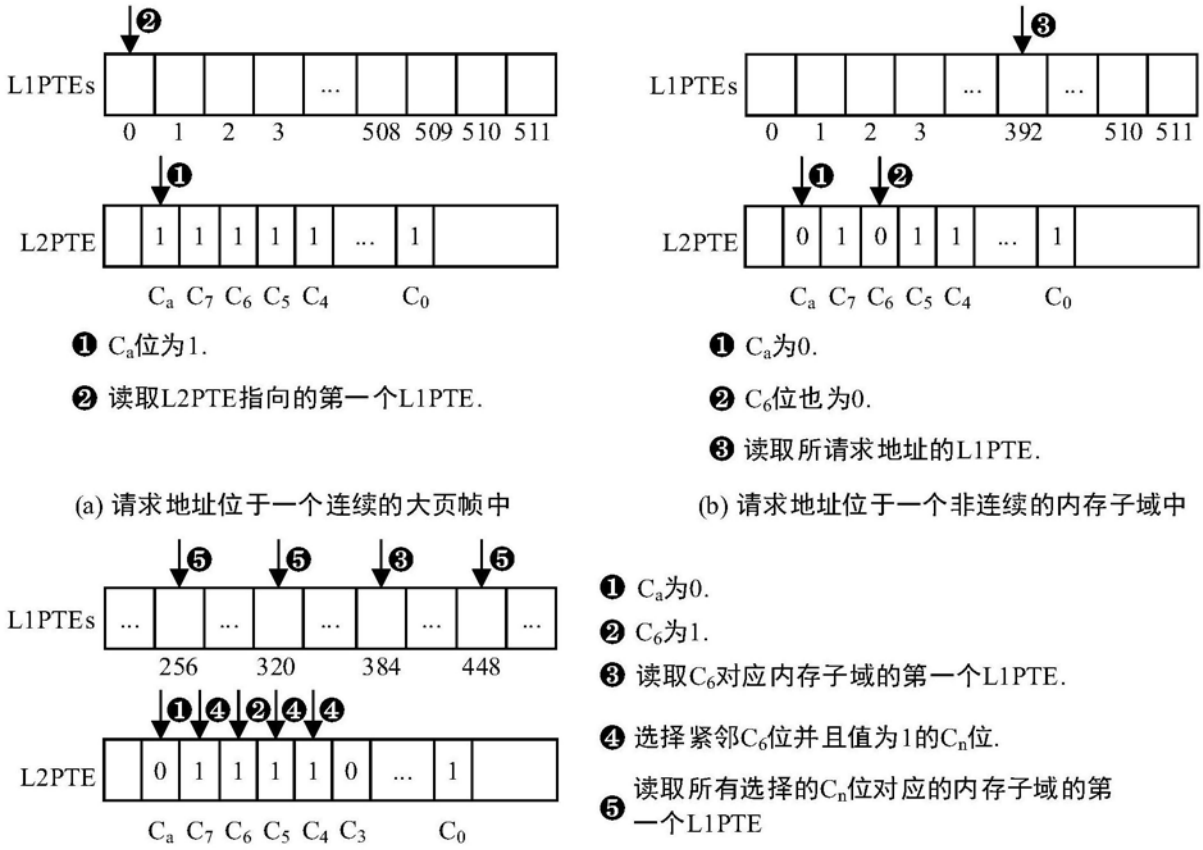

L1PTEs
0 1 2 3 ... 508 509 510 511
L2PTE
1 1 1 1 1 ... 1
Ca C7 C6 C5 C4 C0
❶ Ca位为1.
❷ 读取L2PTE指向的第一个L1PTE.
(a) 请求地址位于一个连续的大页帧中
L1PTEs
0 1 2 3 392 510 511
L2PTE
0 1 0 1 1 ... 1
Ca C7 C6 C5 C4 C0
❶ Ca为0.
❷ C6位也为0.
❸ 读取所请求地址的L1PTE.
(b) 请求地址位于一个非连续的内存子域中
L1PTEs
256 320 384 448
L2PTE
0 1 1 1 1 0 ... 1
Ca C7 C6 C5 C4 C3 C0
❶ Ca为0.
❷ C6为1.
❸ 读取C6对应内存子域的第一个L1PTE.
❹ 选择紧邻C6位并且值为1的Cn位.
❺ 读取所有选择的Cn位对应的内存子域的第一个L1PTE
(c) 请求地址位于一个连续的内存子域中

图3

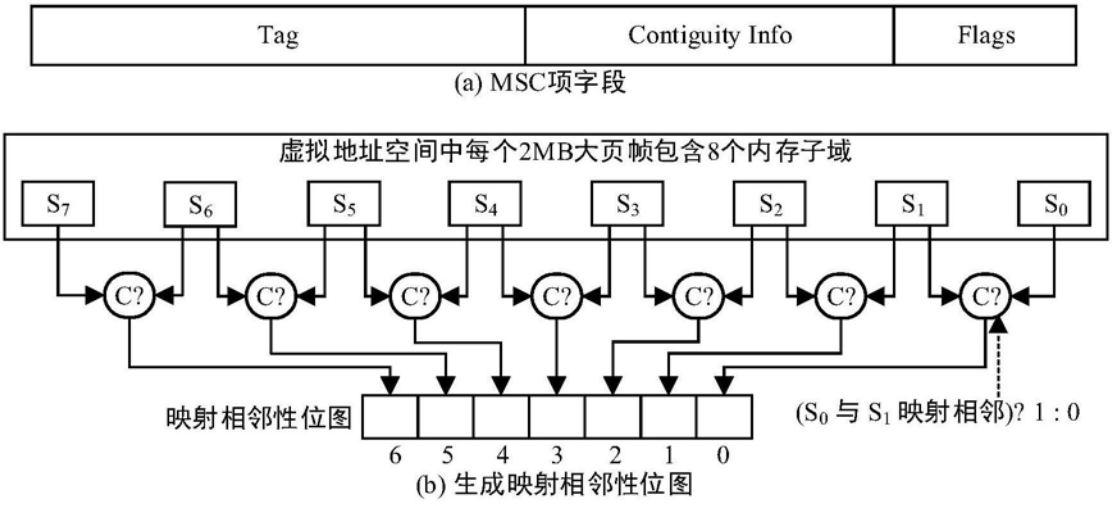

Tag
Contiguity Info
Flags
(a) MSC项字段
虚拟地址空间中每个2MB大页帧包含8个内存子域
S7 S6 S5 S4 S3 S2 S1 S0
C?
映射相邻性位图
6 5 4 3 2 1 0
(S0 与 S1 映射相邻)? 1 : 0
(b) 生成映射相邻性位图

图4

图5