

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2017年11月9日 (09.11.2017)



(10) 国际公布号
WO 2017/190266 A1

- (51) 国际专利分类号:
G06F 12/00 (2006.01)
- (21) 国际申请号: PCT/CN2016/080867
- (22) 国际申请日: 2016年5月3日 (03.05.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (72) 发明人: 方磊 (FANG, Lei); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 蔡卫光 (CAI, Weiguang); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 顾雄礼 (GU, Xiongli); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (74) 代理人: 北京龙双利达知识产权代理有限公司 (LONGSUN LEAD IP LTD.); 中国北京市海淀区北清路68号院3号楼101, Beijing 100094 (CN)。
- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH,

(54) Title: METHOD FOR MANAGING TRANSLATION LOOKASIDE BUFFER AND MULTI-CORE PROCESSOR

(54) 发明名称: 管理转址旁路缓存的方法和多核处理器

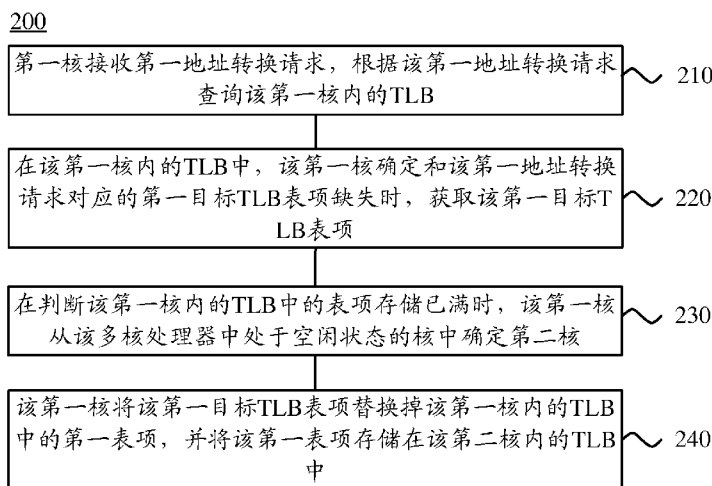


图 2

- 210 A first core receives a first address translation request, and queries a TLB in the first core according to the first address translation request
- 220 In the TLB of the first core, the first core acquires a first target TLB table entry upon determining that the first target TLB table entry corresponding to the first address translation request is missing
- 230 Upon determining that a table entry storage of the TLB in the first core is full, the first core determines a second core from idle cores of the multi-core processor
- 240 The first core replaces a first table entry of the TLB in the first core with the first target TLB table entry, and stores the first table entry in a TLB of the second core

(57) Abstract: A method for managing a translation lookaside buffer (TLB) and a multi-core processor, the method being applied to the multi-core processor. The multi-core processor comprises a first core, and the first core comprises a TLB. The method comprises: the first core receiving a first address translation request, and querying the TLB in the first core according to the first address translation request (210); in the TLB of the first core, acquiring a first target TLB table entry upon determining that the first target TLB table entry corresponding to the first address translation request is missing (220); determining a second core from idle cores of the multi-core processor upon determining a table entry storage of the TLB in the first core is full (230); replacing a first table entry of the TLB in



WO 2017/190266 A1

TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

根据细则4.17的声明:

- 关于申请人有权申请并被授予专利(细则4.17(ii))

本国际公布:

- 包括国际检索报告(条约第21条(3))。

the first core with the first target TLB table entry, and storing the first table entry in a TLB of the second core (240). By utilizing the TLB resources of the idle cores, the method can increase the TLB capacity of the operating cores, reduce the ratio of missing TLB entries and speed up the execution of processes.

(57) 摘要: 一种管理转址旁路缓存TLB的方法和多核处理器, 其方法用于多核处理器, 多核处理器包括第一核, 第一核包含一个TLB, 该方法包括: 第一核接收第一地址转换请求, 根据第一地址转换请求查询第一核内的TLB(210); 在第一核内的TLB中, 确定和第一地址转换请求对应的第一目标TLB表项缺失时, 获取第一目标TLB表项(220); 在判断第一核内的TLB中的表项存储已满时, 从多核处理器中处于空闲状态的核中确定第二核(230); 将第一目标TLB表项替换掉第一核内的TLB中的第一表项, 并将第一表项存储在第二核内的TLB中(240)。本方法通过利用空闲状态的核的TLB资源来扩大工作的核的TLB容量, 能够降低TLB缺失率, 加快程序的执行。

管理转址旁路缓存的方法和多核处理器

技术领域

本发明涉及信息技术领域，并且更具体地，涉及管理转址旁路缓存
5 (Translation Lookaside Buffer, TLB) 的方法和多核处理器。

背景技术

用户程序一般运行在虚拟地址空间中，在用户程序执行过程中，由操作系统 (Operating System, OS) 和内存管理单元 (Memory Management Unit, MMU) 负责将访存请求携带的虚拟地址转换成对应物理内存空间的物理地址。虚拟地址包括虚拟页号 (Virtual Page Number, VPN) 和页内偏移，物理地址包括物理框号 (Physical Frame Number, PFN) 和页内偏移，在虚拟地址转换为物理地址的过程中，虚拟地址中的页内偏移和物理地址中的页内偏移保持不变，而是通过映射关系将虚拟页号转换为物理框号。一般情况下，
15 映射关系是以表项的方式存储在内存上的页表 (Page Table) 中。一次地址转换需要经过操作系统处理和内存中页表的访问，会产生较大的时延及性能损失，所以片上多处理器 (Chip Multi-Processor, CMP) (也可以称为多核处理器) 中的每个处理器核 (如下简称核 (Core)) 会存储一个转址旁路缓存 TLB, TLB 中保存了一些 VPN 到 PFN 的转换表项。

20 随着技术的进步，现有多核场景下，系统中的程序工作集持续增大，也即程序和待处理数据越来越多，然而由于核中的 TLB 的存储空间有限，因此，随着应用程序和待处理数据的增加，现有核中的 TLB 存储的表项越来越不能满足使用需求，导致 TLB 中可能缺失 (即: TLB Miss) 当前需要的 TLB 转换表项，即导致 TLB 的缺失率增大。若当前需要的 TLB 转换表项缺
25 失，该核通常需要从内存中经过操作系统处理和内存中页表的访问，来获取该 TLB 转换表项，这样会产生较大的时延及性能损失，降低了程序的执行效率。

因此，如何降低 TLB 缺失率，加快程序的执行，成为亟待解决的问题。

30 发明内容

本发明实施例提供一种处理转址旁路缓存的方法和多核处理器，该方法

能够扩大处于工作状态的核内的 TLB 容量，从而降低 TLB 缺失率，加快程序的执行。

第一方面，提供了一种管理转址旁路缓存 TLB 的方法，应用于多核处理器，该多核处理器包括第一核，该第一核包含一个 TLB，该方法包括：

5 该第一核接收第一地址转换请求，根据该第一地址转换请求查询该第一核内的 TLB。

在该第一核内的 TLB 中，该第一核确定和该第一地址转换请求对应的第一目标 TLB 表项缺失时，获取该第一目标 TLB 表项。

10 在判断该第一核内的 TLB 中的表项存储已满时，该第一核从该多核处理器中处于空闲状态的核中确定第二核。

该第一核将该第一目标 TLB 表项替换掉该第一核内的 TLB 中的第一表项，并将该第一表项存储在该第二核内的 TLB 中。

具体地，该第一核可以从多核处理器中的其他核中获得该第一目标 TLB 表项，例如，第一核可以向多核系统中的其他核广播 TLB 查询请求，
15 并在该广播的 TLB 查询请求中携带导致缺失的虚拟地址，即第一目标 TLB 表项对应的虚拟地址，其他的核收到广播地址后，在本地的 TLB 中查找此虚拟地址，如果其中一个处理器核内的 TLB 命中，则可以将该第一目标 TLB 表项反馈给该第一核。

20 通过这种方式，使得第一核可以从其他的核中快速的获的该第一目标 TLB 表项，避免该第一核向操作系统发起查询请求，从内存中获取该第一目标 TLB 表项，能够节省一定的时间，提高了应用效率。

25 另外，第一核也可以是从内存的页表中获得该第一目标 TLB 表项，例如，第一核向操作系统发送查询请求，查询请求中携带导致缺失的虚拟地址，经过操作系统处理从内存的页表中获得该第一目标 TLB 表项。本发明实施例并不限于此。

也就是说，在第一核（也可以称为工作的核或工作节点）发生 TLB 缺失（Miss），并且本地 TLB 中的表项存储已满时，为了存储获取的第一目标 TLB 表项，该第一核需要用第一目标 TLB 表项替换一个有效 TLB 表项。在这种情况下，第一核试图获取更多 TLB 资源，来保存被替换下的 TLB 表项，
30 因此，第一核需要从空闲核中确定第二核。

因此，本发明实施例通过使用空闲核内的 LTB 资源来保存被替换下的

TLB 表项，不但提高了空闲核 TLB 资源的利用率，还间接的提高了第一核的 TLB 的容量，减低了第一核通过内存来获取目标 TLB 表项的发生，加快程序的执行。

5 应理解，本发明实施例中的多核处理包括的核也可以称为节点，本文中的节点可以与多核处理器中的核等同。

例如，该第一核将第一目标 TLB 表项替换掉 该第一核的 TLB 中的第一表项位置中的第一表项，并将该第一表项存储在第二核的 TLB 中。

10 应理解，第一表项位置可以为主核的 TLB 中任意一个表项位置，本发明实施例并不对此做限定。换句话说，第一表项可以为第一核的 TLB 中的任意一个 TLB 表项，例如，第一表项可以是第一核的 TLB 中的第一个表项、最后一个表项或者中间的一个表项，本发明实施例并不对此做限定。

也就是说，在第一核的 TLB 表项已存储满，且第一目标 TLB 表项缺失时，该第一核需要将第一目标 TLB 表项存储在该第一核的 TLB 的第一表项位置，并将由第一表项位置替换下来的第一表项存储在第二核的 TLB 中。

15 在本发明实施例中，可以将第一核称为第二核的主核，可以将第二核称为第一核的从核（或备用核）。

20 应理解，主核（Master）将 TLB 表项写入从核（Slave）的 TLB 中。这个过程发生在 Master 中的 TLB 表项被替换的情况下，当 Master 中获得第一目标 TLB 表项后，由于主核的 TLB 表项已存储满，因此，需要将该第一目标 TLB 表项重填到第一表项位置，也即用第一目标 TLB 表项将第一表项位置中的第一表项替换下来，然后将第一表项存储在从核（Slave）（例如，第二核）的 TLB 中。

25 其中，在将被替换表项写入 Slave 的 TLB 的过程中，由 Slave 来保存写入位置。本发明实施例中可以采用轮流写入机制，即如果从核的 TLB 表项数为 N，则从第一个表项开始，即从 0 开始依次写入直到 N-1 个表项，即直到表项全满，在从核（例如，第二核）的表项存储已满时的 TLB 的处理方法，将在下文中详细描述。

30 因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项后替换第一核的 TLB 中的第一表项，并将第一表项存储在第二核的 TLB 中，由于将替换下来的表项存储在了第二核的 TLB 中。因此，本发明实施例通过利用空闲核的 TLB 资源来扩

大工作的核的 TLB 容量，从而能够降低 TLB 缺失率，加快程序的执行。

由于工作的核即主核 (Master) (例如，第一核) 将被替换的 TLB 表项写入空闲节点即从核 (Slave) (例如，第二核) 的 TLB 中。当工作的核需要再次用到这些被替换的 TLB 表项时，不需要通过操作系统来重新获取这些表项，通过直接访问 Slave 的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

应注意，本发明实施例中一个空闲的核仅能将自身的 TLB 资源共享给一个工作的核使用；一个工作的核可以获取多核空闲的核的 TLB 资源存储 TLB 表项。

10 进一步地，在第一方面的一种实现方式中，该第一核从该多核处理器中处于空闲状态的核中确定第二核，包括：

该第一核向该多核处理器中其他核中每个核发送状态查询请求，该状态查询请求用于查询该每个核是否处于空闲状态；

15 该第一核接收该其他核中每个核发送的响应消息，该响应消息用于指示该每个核是否处于空闲状态；

该第一核根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核。

20 因此，本发明实施例通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，还间接的提高了第一核的 TLB 的容量，减低了第一核通过内存来获取目标 TLB 表项的发生，加快程序的执行。

进一步地。在第一方面的另一种实现方式中，该第一核根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核，包括：

25 该第一核根据该响应消息确定空闲核列表，该空闲核列表中包括该多核处理器中除该第一核外的其他核中处于空闲状态的核；

在该空闲核列表中，选择与该第一核的通信开销最小的空闲核作为该第二核。

30 因此，本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核，并将替换下的 TLB 表项存储在第二核中，最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时，由于通信开销很小，所以第一核能够快速的查询到该 TLB 表项，提高程序的执行效率。

本领域的技术人员可以理解，基于与第一核的通信开销最小作为选择第二核的原则是一种较为理想的选择方式，在具体实现中，通信开销的判断需要结合多处理器芯片中的片上网络 NoC (Network on Chip) 路由的拥塞程度来决定，基于网络拥塞程度从空闲核中选择某一处理器核作为第二核，为了简化第二核的选择过程，可采用如下方面的实现方式：

进一步地，在第一方面的另一种实现方式中，在该空闲核列表中，选择与该第一核的通信开销最小的空闲核作为该第二核，包括：

该第一核将该空闲核列表中与该第一核的通信跳数最小的空闲核作为该第二核；或者，

10 该第一核将该空闲核列表中与该第一核的物理距离最小的空闲核作为该第二核。

其中，本发明实施例中还可以采用其他方式确定通信开销最小的核，本发明实施例并不限于此。

例如，第一核（这里可以称为请求节点）向多核处理器中其他核（也可以称为其他节点）广播状态查询请求（也可以称为 TLB 查询请求）；该状态查询请求用于查询该每个核是否处于空闲状态；所每个核收到状态查询请求之后，会向第一核（请求节点）发送响应消息，该响应消息用于指示该核是否处于空闲状态。这样第一核根据响应消息就获得了空闲核列表。若空闲核列表为空，则结束对 TLB 资源的获取，则按照现有的方式读取内存中的该第一核缺失的 TLB 表项（即第一目标 TLB 表项）。若空闲核列表非空，则进行 TLB 资源的获取，即第一核根据和空闲核列表中空闲的核之间的通信开销，选取通信开销最小的空闲核，向其发送 TLB 共享请求。若此时这个空闲核已经被其他节点共享，或其变为工作状态，则向请求节点发送失败反馈，将此节点从空闲节点列表中移除，然后重复上述过程。若此时这个空闲核是空闲状态，则确定该空闲核为第二核。

因此，本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核，并将替换下的 TLB 表项存储在第二核中，最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时，由于通信开销很小，所以第一核能够快速查询到该 TLB 表项，提高程序的执行效率。

30 可选地，在第一方面的另一种实现方式中，在确定该第二核之后，该方法还包括：

将该第二核的标识记录在该第一核内的 TLB 备用核列表中。

这里备用核列表也可以称为从核列表。

具体而言，在第一核从空闲核列表中确定出第二核后，第二核成为第一核的从核 (Slave) (也可以称为备用核)。该从核 (第二核) 将第一核的标识 (例如，编号等) 写入该从核的主核 (Master) 编号寄存器中，第一核成为第二核的主核 (Master)，并将该从核 (第二核) 的标识 (例如编号等) 加入其备用核列表中。

因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项确定第二核，并将第二核记录在该第一核的备用核列表中，这样第一核根据备用核列表可以对备用核列表中的第二核内的 TLB 进行读写，进而提升第一核的 TLB 的容量，从而能够降低第一核的 TLB 缺失率，加快程序的执行。

可选地，在第一方面的另一种实现方式中，该方法还包括：

在该第一核从该多核处理器处于空闲状态的核中确定第二核之后，该方法还包括：

该第一核接收第二地址转换请求，根据该第二地址转换请求查询该第一核内的 TLB；

在该第一核内的 TLB 中，该第一核确定和该第二地址转换请求对应的第二目标 TLB 表项缺失时，查询该第二核内的 TLB；

当在该第二核内的 TLB 中查询到该第二目录 TLB 表项时，将该第二目标 TLB 表项替换掉该第一核内的 TLB 中的第二表项，并将该第二表项存储在该第二核内的 TLB 中。

例如，第一核将第二表项存储在第二核中原第二目标 TLB 表项的位置。也就是说，第一核将第二表项与第二目标 TLB 表项互换存储位置。

也就是说，当主核本地的 TLB 表项缺失时，主核会读取从核中的 TLB 表项。

上述过程发生在 Master (第一核) 本地 TLB 缺失 (Miss)，即本地 TLB 不存在第二目标 TLB 表项的情况下，Master 会读取 Slave 中的 TLB 表项。Master 根据备用核列表 (Slave List) (也称为从核列表)，向所有的 Slave 发送 TLB 读取请求。Slave 接收到 TLB 读取请求之后，查询从核本地的 TLB，如果第二目标 TLB Miss，则返回缺失反馈；如果命中，则返回命中反馈，

以及命中 TLB 的表项内容。Master 收集所有反馈之后，若都是缺失反馈，则向操作系统发送 TLB 缺失请求；若有 Slave 命中，则利用命中 TLB 表项进行重填。当在重填过程若发生表项替换的话，则将被替换表项写到命中的从核中。

5 因此，本发明实施例中工作的核拥有空闲的核的 TLB 的读/写权限，通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，更进一步提高了第一核的 TLB 的容量，在第一核内的 TLB 缺失表项时，可以通过读取从核内的 TLB 获取目标表项，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

10 在 Master 中的有效 TLB 表项被替换，并且备用核列表中的所有 Slave 都是 Full Slave 的情况时，由于 Full Slave 的 TLB 中保存的都是 Master 的被替换表项，这些表项后续可能会被 Master 用到，因此，Master 希望获取更多的 Slave 来保存被替换表项。

15 相应地，在第一方面的另一种实现方式中，该方法还包括：

 该第一核接收第三地址转换请求，根据该第三地址转换请求查询该第一核内的 TLB；

 在该第一核内的 TLB 中，该第一核确定和该第三地址转换请求对应的第三目标 TLB 表项缺失时，查询该第二核内的 TLB；

20 当在该第二核内的 TLB 中确定第三目标 TLB 表项缺失时，获取该第三目标 TLB 表项；

 在判断该第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时，该第一核从该多核处理器处于空闲状态的核中确定第三核；

25 该第一核将该第三目标 TLB 表项替换掉该第一核内的 TLB 中的第三表项，并将该第三表项存储在该第三核内的 TLB 中。

 具体而言，在主核（第一核）的 TLB 中的表项已存储满，且该主核的所有从核（例如这里只有一个从核，为第二核）中的 TLB 表项已存储时，该主核（第一核）从该主核从多核处理器中的其他空闲核中确定第三核用于存储被第三目标 TLB 表项替换下来的第一核的 TLB 中的第三表项。

30 因此，本发明实施例在当前的从核存储已全满后，会获取新的从核来保存替换下的 TLB 表项，进一步扩大了该第一核的容量，当该第一核再次查

询该替换下来的 TLB 表项时, 该第一核可以直接从该新的从核中读取, 无需该核通过内存来获取。因此, 本发明实施例能够加快程序的执行, 提高程序的执行效率。

5 可选地, 在第一方面的另一种实现方式中, 当该第一核由工作状态转换到空闲状态后, 该方法还包括:

该第一核向该 TLB 备用核列表中记录的核发送 TLB 释放指令, 该 TLB 释放指令用于指示该备用核列表中记录的核解除 TLB 共享。

10 也就是说, Master 变成空闲状态: 释放所有已经获取的 TLB 资源。Master 根据 Slave List 向所有 Slave 发送 TLB 释放请求, 这样这些空闲的核的 TLB 资源可以被其他工作的核使用。Master 释放所有已经获取的 TLB 资源后, 该主核和从核均成为空闲的核, 可以被其他工作的核使用。

15 因此, 本发明实施例在第一核转换到空闲状态后, 通过向所有的从核发送释放指令释放已获取的从核的 TLB 资源, 通过这种方式, 使得这些从核 (空闲的核) 的 TLB 资源释放, 避免资源的浪费, 进而该释放的核的 TLB 资源能够被其他的工作的核使用, 提升其他工作核的容量, 加快工作的核的程序执行。

20 其中, 从核解除 TLB 共享后, 该从核可以将自身存储的 TLB 表项删除, 也可以不删除, 本发明实施例并不对此做限定。例如, 在解出共享后, 在该从核又作为另一个工作的核的从核后, 可以将 TLB 中的表项全部删除。以供该另一个工作的核存储替换下的表项。在例如, 在解除共享后, 在该从核成为工作状态时, 可以保留之前存储的 TLB 表项, 以供自身查找使用。

可选地, 在第一方面的另一种实现方式中, 该方法还包括:

当该第二核由空闲状态转换到工作状态后, 该方法还包括:

25 该第一核接收该第二核发送的 TLB 解除共享请求, 该 TLB 解除共享请求中携带有该第二核的标识;

在该 TLB 备用核列表中, 该第一核将该第二核的标识删除。

30 因此, 本发明实施例在第二核由空闲状态变为工作状态后, 该第二核会向第一核发送 TLB 解除共享请求, 进而使得第一核释放该第二核, 使得第二核能够使用自身的 TLB 资源, 避免了影响第二核正在的业务处理, 同时, 第二核还可以成为另一个主核, 可以使用其他的空闲核的 TLB 资源。

进一步地, 在第一方面的另一种实现方式中, 在该 TLB 备用核列表中,

该第一核将该第二核的标识删除之前，该方法还包括：

该第一核从该多核处理器处于空闲状态的核中确定第四核；

该第一核将该第二核的 TLB 中的所有表项拷贝到该第四核的 TLB 中。

也就是说，在主核（例如，第一核）从备用核列表中删除从核（例如，第二核）之前，会从多核处理器中重新获取一个核（新的从核）（例如，第四核），该新的从核的 TLB 用于存储删除的从核（即第二核）的 TLB 中所有的表项。这样在后续查询中如果需要查询第二核中的表项，不需要通过操作系统来重新获取这些表项，通过直接访问从核（第四核）的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

10 因此，本发明实施例通过第一核获取新的从核（第四核），来保存释放的第二核内的 TLB 中的所有表项，在该第一核在后续查询中如果再次需要查询第二核中的表项，不需要通过操作系统从内存中重新获取这些表项，而可以直接通过访问从核（第四核）的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

15 第二方面，提供了一种多核处理器，该多核处理器能够实现第一方面及其实现方式中的任一实现方式，该多核处理器包括第一核，该第一核内包括一个转址旁路缓存 TLB，

该第一核用于接收第一地址转换请求，根据该第一地址转换请求查询该第一核内的 TLB，在该第一核内的 TLB 中，确定和该第一地址转换请求对
20 应的第一目标 TLB 表项缺失时，获取该第一目标 TLB 表项，在判断该第一核内的 TLB 中的表项存储已满时，从该多核处理器中处于空闲状态的核中确定第二核，将该第一目标 TLB 表项替换掉该第一核内的 TLB 中的第一表项；

该第二核用于将该第一表项存储在该第二核内的 TLB 中。

25 因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项后替换第一核的 TLB 中的第一表项，并将第一表项存储在第二核的 TLB 中，由于将替换下来的表项存储在了第二核的 TLB 中。因此，本发明实施例通过利用空闲核的 TLB 资源来扩大工作的核的 TLB 容量，从而能够降低 TLB 缺失率，加快程序的执行。

30 由于工作的核即主核（Master）（例如，第一核）将被替换的 TLB 表项写入空闲节点即从核（Slave）（例如，第二核）的 TLB 中。当工作的核需要

再次用到这些被替换的 TLB 表项时，不需要通过操作系统来重新获取这些表项，通过直接访问 Slave 的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

5 应注意，本发明实施例中一个空闲的核仅能将自身的 TLB 资源共享给一个工作的核使用；一个工作的核可以获取多核空闲的核的 TLB 资源存储 TLB 表项。

可选地，在第二方面的另一种实现方式中，在获取该第一目标 TLB 表项时，该第一核 1110 具体用于：

从内存的页表中获取该第一目标 TLB 表项，

10 或者，从该多核系统中的其他核获取该第一目标 TLB 表项。

例如，第一核可以向多核系统中的其他核广播 TLB 查询请求，并在该广播的 TLB 查询请求中携带导致缺失的虚拟地址，即第一目标 TLB 表项对应的虚拟地址，其他的核收到广播地址后，在本地的 TLB 中查找此虚拟地址，如果其中一个处理器核内的 TLB 命中，则可以将该第一目标 TLB 表项
15 反馈给该第一核。

通过这种方式，使得第一核可以从其他的核中快速的获的该第一目标 TLB 表项，避免该第一核向操作系统发起查询请求，从内存中获取该第一目标 TLB 表项，能够节省一定的时间，提高了应用效率。

另外，第一核也可以是从内存的页表中获得该第一目标 TLB 表项，例如，
20 第一核向操作系统发送查询请求，查询请求中携带导致缺失的虚拟地址，经过操作系统处理从内存的页表中获得该第一目标 TLB 表项。本发明实施例并不限于此。

进一步地，第二方面的另一种实现方式中，在从该多核处理器中处于空闲状态的核中确定第二核时，该第一核具体用于：

25 向该多核处理器中其他核中每个核发送状态查询请求，该状态查询请求用于查询该每个核是否处于空闲状态；

接收该每个核发送的响应消息，该响应消息用于指示该每个核是否处于空闲状态；

并根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核。

30 因此，本发明实施例通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，还间接的提高了第一核

的 TLB 的容量，减低了第一核通过内存来获取目标 TLB 表项的发生，加快程序的执行。

进一步地，第二方面的另一种实现方式中，在根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核时，该第一核具体用于：

5 根据该响应消息确定空闲核列表，该空闲核列表中包括该多核处理器中除该第一核外的其他核中处于空闲状态的核；

在该空闲核列表中，选择与该第一核的通信开销最小的空闲核作为该第二核。

10 因此，本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核，并将替换下的 TLB 表项存储在第二核中，最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时，由于通信开销很小，所以第一核能够快速查询到该 TLB 表项，提高程序的执行效率。

进一步地，第二方面的另一种实现方式中，在选择与该第一核的通信开销最小的空闲核作为该第二核时，该第一核具体用于：

15 将该空闲核列表中与该第一核的通信跳数最小的空闲核作为该第二核；或者，

将该空闲核列表中与该第一核的物理距离最小的空闲核作为该第二核。

可选地，第二方面的另一种实现方式中，在确定该第二核之后，该第一核还用于将该第二核的标识记录在该第一核内的 TLB 备用核列表中。

20 因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项确定第二核，并将第二核记录在该第一核的备用核列表中，这样第一核根据备用核列表可以对备用核列表中的第二核内的 TLB 进行读写，进而提升第一核的 TLB 的容量，从而能够降低第一核的 TLB 缺失率，加快程序的执行。

25 可选地，第二方面的另一种实现方式中，该第一核还用于：

接收第二地址转换请求，根据该第二地址转换请求查询该第一核内的 TLB；

在该第一核内的 TLB 中，确定和该第二地址转换请求对应的第二目标 TLB 表项缺失时，查询该第二核内的 TLB；

30 当在该第二核内的 TLB 中查询到该第二目录 TLB 表项时，将该第二目标 TLB 表项替换掉该第一核内的 TLB 中的第二表项，

该第二核还用于将该第二表项存储在该第二核内的 TLB 中。

因此，本发明实施例中工作的核拥有空闲的核的 TLB 的读/写权限，通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，更进一步提高了第一核的 TLB 的容量，在第一核内的 TLB 缺失表项时，可以通过读取从核内的 TLB 获取目标表项，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

可选地，第二方面的另一种实现方式中，该第一核还用于：

接收第三地址转换请求，根据该第三地址转换请求查询该第一核内的 TLB；

在该第一核内的 TLB 中，确定和该第三地址转换请求对应的第三目标 TLB 表项缺失时，查询该第二核内的 TLB；

当在该第二核内的 TLB 中确定第三目标 TLB 表项缺失时，获取该第三目标 TLB 表项；

在判断该第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时，从该多核处理器处于空闲状态的核中确定第三核；

将该第三目标 TLB 表项替换掉该第一核内的 TLB 中的第三表项，

该第三核用于将该第三表项存储在该第三核内的 TLB 中。

因此，本发明实施例在当前的从核存储已全满后，会获取新的从核来保存替换下的 TLB 表项，进一步扩大了该第一核的容量，当该第一核再次查询该替换下来的 TLB 表项时，该第一核可以直接从该新的从核中读取，无需该核通过内存来获取。因此，本发明实施例能够加快程序的执行，提高程序的执行效率。

可选地，第二方面的另一种实现方式中，当该第一核由工作状态转换到空闲状态后，该第一核还用于：

向该 TLB 备用核列表中记录的核发送 TLB 释放指令，该 TLB 释放指令用于指示该备用核列表中记录的核解除 TLB 共享。

因此，本发明实施例在第一核转换到空闲状态后，通过向所有的从核发送释放指令释放已获取的从核的 TLB 资源，通过这种方式，使得这些从核（空闲的核）的 TLB 资源释放，避免资源的浪费，进而该释放的核的 TLB 资源能够被其他的工作的核使用，提升其他工作核的容量，加快工作的核的

程序执行。

可选地，第二方面的另一种实现方式中，当该第二核由空闲状态转换到工作状态后，

所第二核用于向该第一核发送的 TLB 解除共享请求，该 TLB 解除共享请求中携带有该第二核的标识；

该第一核还用于接收该 TLB 解除共享请求，在该 TLB 备用核列表中，将该第二核的标识删除。

因此，本发明实施例在第二核由空闲状态变为工作状态后，该第二核会向第一核发送 TLB 解除共享请求，进而使得第一核释放该第二核，使得第二核能够使用自身的 TLB 资源，避免了影响第二核正在的业务处理，同时，第二核还可以成为另一个主核，可以使用其他的空闲核的 TLB 资源。

可选地，第二方面的另一种实现方式中，该第一核还用于从该多核处理器处于空闲状态的核中确定第四核；

该第四核的 TLB 用于存储该第二核的 TLB 中的所有表项。

因此，本发明实施例通过第一核获取新的从核（第四核），来保存释放的第二核内的 TLB 中的所有表项，在该第一核后续查询中如果再次需要查询第二核中的表项，不需要通过操作系统从内存中重新获取这些表项，而可以直接通过访问从核（第四核）的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

20

附图说明

为了更清楚地说明本发明实施例的技术方案，下面将对本发明实施例中所需要使用的附图作简单地介绍。

图 1 是根据本发明一个实施例的多核处理器的结构示意图。

25 图 2 是根据本发明一个实施例的管理转址旁路缓存 TLB 的方法的示意图。

图 3 是根据本发明一个实施例的备用核列表向量的示意图。

图 4 是根据本发明另一实施例的备用核列表向量的示意图。

30 图 5 是根据本发明一个实施例的管理转址旁路缓存 TLB 的方法的示意图。

图 6 是根据本发明另一实施例的管理转址旁路缓存 TLB 的方法的示意图。

图。

图 7 是根据本发明另一实施例的管理转址旁路缓存 TLB 的方法的示意图。

5 图 8 是根据本发明另一实施例的管理转址旁路缓存 TLB 的方法的示意图。

图 9 是根据本发明另一实施例的管理转址旁路缓存 TLB 的方法的示意图。

图 10 是根据本发明另一实施例的管理转址旁路缓存 TLB 的方法的示意图。

10 图 11 是根据本发明一个实施例的多核处理器示意框图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明的一部分实施例，而不是全部实施例。

15 本发明的技术方案，可以运行在包括例如，CPU、存储器管理单元（MMU，Memory Management Unit）、内存的硬件设备上，该硬件设备所运行的操作系统可以是各种通过线程或进程（包括多个线程）实现业务处理的操作系统，例如，Linux 系统、Unix 系统、Windows 系统、Android 系统、
20 OS 系统等。

应理解，以上列举的实时操作系统仅是示例性说明，本发明并未特别限定，只要本发明实施例的硬件设备具有多核处理器即可，本发明实施例并不限于此。

25 为了方便理解本发明实施例，首先在此对本发明实施例描述中的一些术语定义如下：

在本发明实施例中，术语“多核处理器”指的是包含了多个处理器核（Core）的处理器，具体可以表现为片上多核处理器，或者板上多核处理系统。其中，片上多核处理器是多个处理器核（Core）通过片上网络（Network On Chip, NOC）将各个处理器核互连并集成在一个芯片（Chip）上的处理器（Processor），板上多核处理系统指的是多个处理器核中每个核分别封装
30 （Package）为处理器，并集成在电路板上所构成的处理系统。

在本发明实施例中，术语“处理器核”是“处理器核心”的简称，又称为内核或核，是 CPU（Central Processing Unit）最重要的组成部分，它是由单晶硅以一定的生产工艺制造出来的，CPU 所有的计算、接收命令或存储命令、处理数据都由处理器核执行。术语中“多处理器核”指的是包含至少两个处理器核，“多处理器核”涵盖了现有技术中的多核（Multi-Core），以及众核（Many Core）所应用的范围。

在本发明实施例中，TLB 也可以称为页表缓冲，里面存放的是一些页表文件即虚拟地址到物理地址的转换表项。TLB 可以用于虚拟地址与物理地址之间的交互，提供一个寻找物理地址的缓存区，能够有效减少内核寻找物理地址所消耗时间。

在本发明实施例中，“主核（Master）”表示处于工作状态且能够使用其他空闲核的 TLB 资源管理 TLB 表项的核。“从核（Slave）”表示处于空闲状态且能为主核共享自身 TLB 资源的核。

在本发明实施例中，“TLB 缺失”表示核的 TLB 中不存在地址转换请求所对应的 TLB 表项。“TLB 命中”表示核中的 TLB 存在地址转换请求所对应的 TLB 表项。

“TLB 替换”表示主核中的一个 TLB 表项和从核中的与地址转换请求所对应的 TLB 表项互换位置。例如，如图 6 所示，将从核中与地址转换请求所对应的 TLB 表项，即图 6 中的从核内的 TLB 中的“命中”TLB 表项与主核的 TLB 中的“替换”TBL 表项互换位置。

还应理解，在本发明实施例中，第一、第二、第三和第四只是为了区分核，而不应该对本发明的保护范围构成任何限定，例如，第一核也可以称为第二核，第二核可以称为第一核等。

应注意，本发明实施例中，需要根据实际情况的变化，第四核可以与第二核或第三核是相同的核，也可以是不同的核，本发明实施例并不对此做限定。

在多核系统中，通常采用较多的处理器核来提高并行处理性能。但是应用程序中的一些部分并行度较低，导致系统中一些处理器核没有任务可以执行，成为空闲的核（也可以称为空闲状态的核或空闲节点）。本发明实施例就是将空闲的核内的 TLB 资源动态地分配给正在执行任务的工作的核（也可以称为工作状态的核或工作节点），扩大工作的核的 TLB 容量，减少 TLB

缺失，最终达到加快程序运行的目的。工作的核可以获得一个或多个空闲的核的 TLB 资源，来满足其 TLB 访问需求。这里获得 TLB 资源的工作的核被称为主核 (Master)，而提供 TLB 资源的空闲的核称为从核 (Slave)。在 TLB 资源的使用过程中，经常访问的 TLB 表项位于主核中，不常访问的 TLB 表项位于从核中。

应理解，本发明实施例中的多核处理包括的“核”还可以称为“节点”，也即本文中的“节点”可以与多核处理器中包括的“核”等同。本发明实施例中的多核处理器包括多个核，也可以称为多核处理器包括多个节点，本发明实施例并不对此做限定。

10 应理解本法发明实施例中的多核处理器可以包括至少两个核。例如可以包括 2 个核、4 个核、8 个核、16 个核、32 个核等，本发明实施例并不限于此。下面结合图 1 描述本发明实施例的多核处理器的基本结构。

如图 1 所示的多核处理器包括 16 个核 (Core)，分别为 Core0-Core15，每个核包括：

15 处理模块；

缓存模块，例如缓存模块包括一级缓存 (L1) 和二级缓存 (L2)；
和片上网络接口 (Switch)。

每个核中的处理模块内包含有一个 TLB。其中，各个核之间通过片上网络连接，通过片上网络接口互相通信。水平或垂直相邻的两个核通过之间的
20 链路通信，可以称为一跳。例如，Core1 与 Core3 通信的通信路径至少需要三跳，即 Core1-Core2-Core3。

下面将结合图 2 至图 10 详细描述本发明实施例的管理 TLB 的方法。

图 2 是根据本发明一个实施的管理 TLB 的方法的示意图。图 2 所示的方法可以由第一核执行。具体地，图 2 所示的方法 200 应用于多核处理
25 器，该多核处理器包括第一核，该第一核包含一个 TLB，应理解，该第一核可以是该多核处理器中的任意一个核，例如以图 1 所示的多核处理器 100 而言，第一核可以是 Core0-Core15 中的任意一个，本发明实施例并不对此做限定。具体地，图 2 所示的方法 200 包括：

30 210，第一核接收第一地址转换请求，根据该第一地址转换请求查询该第一核内的 TLB；

换句话说，第一核接收到第一地址转换请求后，查询该第一核内的 TLB

中是否存在与该第一地址转换请求对应的 TLB 表项。

220, 在该第一核内的 TLB 中, 该第一核确定和该第一地址转换请求对应的第一目标 TLB 表项缺失时, 获取该第一目标 TLB 表项;

具体地, 该第一核可以是从小核处理器中的其他核中获得该第一目标 TLB 表项, 例如, 第一核可以向多核系统中的其他核广播 TLB 查询请求, 并在该广播的 TLB 查询请求中携带导致缺失的虚拟地址, 即第一目标 TLB 表项对应的虚拟地址, 其他的核收到广播地址后, 在本地的 TLB 中查找此虚拟地址, 如果其中一个处理器核内的 TLB 命中, 则可以将该第一目标 TLB 表项反馈给该第一核。

10 通过这种方式, 使得第一核可以从其他的核中快速的获的该第一目标 TLB 表项, 避免该第一核向操作系统发起查询请求, 从内存中获取该第一目标 TLB 表项, 能够节省一定的时间, 提高了应用效率。

另外, 第一核也可以是从内存的页表中获得该第一目标 TLB 表项, 例如, 第一核向操作系统发送查询请求, 查询请求中携带导致缺失的虚拟地址, 15 经过操作系统处理从内存的页表中获得该第一目标 TLB 表项。本发明实施例并不限于此。

230, 在判断该第一核内的 TLB 中的表项存储已满时, 该第一核从该多核处理器中处于空闲状态的核中确定第二核。

也就是说, 在第一核 (也可以称为工作的核或工作节点) 发生 TLB 缺失 (Miss), 并且本地 TLB 中的表项存储已满时, 为了存储获取的第一目标 TLB 表项, 该第一核需要用第一目标 TLB 表项替换一个有效 TLB 表项。在这种情况下, 第一核试图获取更多 TLB 资源, 来保存被替换下的 TLB 表项, 因此, 第一核需要从空闲核中确定第二核。

因此, 本发明实施例通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项, 不但提高了空闲核 TLB 资源的利用率, 还间接的提高了第一核的 TLB 的容量, 减低了第一核通过内存来获取目标 TLB 表项的发生, 加快程序的执行。

具体地, 作为另一实施例, 该第一核从该多核处理器中处于空闲状态的核中确定第二核, 包括:

30 该第一核向该多核处理器中其他核中每个核发送状态查询请求, 该状态查询请求用于查询该每个核是否处于空闲状态;

该第一核接收该其他核中每个核发送的响应消息，该响应消息用于指示该每个核是否处于空闲状态；

该第一核根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核。

5 进一步地，该第一核根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核，包括：

该第一核根据该响应消息确定空闲核列表，该空闲核列表中包括该多核处理器中除该第一核外的其他核中处于空闲状态的核；

10 在该空闲核列表中，选择与该第一核的通信开销最小的空闲核作为该第二核。

因此，本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核，并将替换下的 TLB 表项存储在第二核中，最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时，由于通信开销很小，所以第一核能够快速查询到该 TLB 表项，提高程序的执行效率。

15 本领域的技术人员可以理解，基于与第一核的通信开销最小作为选择第二核的原则是一种较为理想的选择方式，在具体实现中，通信开销的判断需要结合多处理器芯片中的片上网络 NoC (Network on Chip) 路由的拥塞程度来决定，基于网络拥塞程度从空闲核中选择某一处理器核作为第二核，为了简化第二核的选择过程，可采用如下的实现方式：

20 (1) 该第一核将该空闲核列表中与该第一核的通信跳数最小的空闲核作为该第二核；或者，

(2) 该第一核将该空闲核列表中与该第一核的物理距离最小的空闲核作为该第二核。

25 其中，本发明实施例中还可以采用其他方式确定通信开销最小的核，本发明实施例并不限于此。

例如，以图 1 的多核处理器为例，针对上述实现方式 (1)，假设第一核为 Core5，空闲核列表中包括的空闲核为 Core7、Core11 和 Core14，由于 Core5 至 Core7 的最少通信跳数为 2 跳，即通信路径为 Core5-Core6-core7；Core5 至 Core11 的最少通信跳数为 3 跳，例如其中一个最少跳数的通信路径为
30 Core5-Core6-core7-Core11；Core5 至 Core14 的最少通信跳数为 3 跳，例如其中一个最少跳数的通信路径为 Core5-Core9-core13-Core14。因此，Core5 选

择与其通信跳数最小的 Core7 作为上述第二核。

针对上述实现方式 (2) 根据图 1 中的多核处理器的分布可知, 由于 Core7、Core11 和 Core14 中与 Core5 的物理距离最小的核为 Core7, 因此, Core5 选择 Core7 作为上述第二核。

5 具体而言, 第一核 (这里可以称为请求节点) 向多核处理器中其他核 (也可以称为其他节点) 广播状态查询请求 (也可以称为 TLB 查询请求); 该状态查询请求用于查询该每个核是否处于空闲状态; 所每个核收到状态查询请求之后, 会向第一核 (请求节点) 发送响应消息, 该响应消息用于指示该核是否处于空闲状态。这样第一核根据响应消息就获得了空闲核列表。若空闲核列表为空, 则结束对 TLB 资源的获取, 则按照现有的方式读取内存中的该第一核缺失的 TLB 表项 (即第一目标 TLB 表项)。若空闲核列表非空, 则进行 TLB 资源的获取, 即第一核根据和空闲核列表中空闲的核之间的通信开销, 选取通信开销最小的空闲核, 向其发送 TLB 共享请求。若此时这个空闲核已经被其他节点共享, 或其变为工作状态, 则向请求节点发送失败反
10 馈, 将此节点从空闲节点列表中移除, 然后重复上述过程。若此时这个空闲核是空闲状态, 则确定该空闲核为第二核。

因此, 本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核, 并将替换下的 TLB 表项存储在第二核中, 最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时, 由于通信开销很小, 所以第一核能够快速查询到该 TLB 表项, 提高程序的执行效率。
20

240, 该第一核将该第一目标 TLB 表项替换掉该第一核内的 TLB 中的第一表项, 并将该第一表项存储在该第二核内的 TLB 中。

例如, 该第一核将第一目标 TLB 表项替换掉该第一核的 TLB 中的第一表项位置中的第一表项, 并将该第一表项存储在第二核的 TLB 中。

25 应理解, 第一表项位置可以为主核的 TLB 中任意一个表项位置, 本发明实施例并不对此做限定。换句话说, 第一表项可以为第一核的 TLB 中的任意一个 TLB 表项, 例如, 第一表项可以是第一核的 TLB 中的第一个表项、最后一个表项或者中间的一个表项, 本发明实施例并不对此做限定。

也就是说, 在第一核的 TLB 表项已存储满, 且第一目标 TLB 表项缺失
30 时, 该第一核需要将第一目标 TLB 表项存储在该第一核的 TLB 的第一表项位置, 并将由第一表项位置替换下来的第一表项存储在第二核的 TLB 中。

在本发明实施例中，可以将第一核称为第二核的主核，可以将第二核称为第一核的从核（或备用核）。

应理解，主核（Master）将 TLB 表项写入从核（Slave）的 TLB 中。这个过程发生在 Master 中的 TLB 表项被替换的情况下，如图 2 所示，当 Master 中获得第一目标 TLB 表项后，由于主核的 TLB 表项已存储满，因此，需要将该第一目标 TLB 表项重填到第一表项位置，也即用第一目标 TLB 表项将第一表项位置中的第一表项替换下来，然后将第一表项存储在从核（Slave）（例如，第二核）的 TLB 中。

其中，在将被替换表项写入 Slave 的 TLB 的过程中，由 Slave 来保存写入位置。本发明实施例中可以采用轮流写入机制，即如果从核的 TLB 表项数为 N，则从第一个表项开始，即从 0 开始依次写入直到 N-1 个表项，即直到表项全满，在从核（例如，第二核）的表项存储已满时的 TLB 的处理方法，将在下文中详细描述。

因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项后替换第一核的 TLB 中的第一表项，并将第一表项存储在第二核的 TLB 中，由于将替换下来的表项存储在了第二核的 TLB 中。因此，本发明实施例通过利用空闲核的 TLB 资源来扩大工作的核的 TLB 容量，从而能够降低 TLB 缺失率，加快程序的执行。

由于工作的核即主核（Master）（例如，第一核）将被替换的 TLB 表项写入空闲节点即从核（Slave）（例如，第二核）的 TLB 中。当工作的核需要再次用到这些被替换的 TLB 表项时，不需要通过操作系统来重新获取这些表项，通过直接访问 Slave 的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

应注意，本发明实施例中一个空闲的核仅能将自身的 TLB 资源共享给一个工作的核使用；一个工作的核可以获取多核空闲的核的 TLB 资源存储 TLB 表项。

可选地，作为另一实施例，在确定该第二核之后，该方法还可以包括：将该第二核的标识记录在该第一核内的 TLB 备用核列表中。

这里备用核列表也可以称为从核列表。

因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项确定第二核，并将第二核记录在

该第一核的备用核列表中，这样第一核根据备用核列表可以对备用核列表中的第二核内的 TLB 进行读写，进而提升第一核的 TLB 的容量，从而能够降低第一核的 TLB 缺失率，加快程序的执行。

在第一核从空闲核列表中确定出第二核后，第二核成为第一核的从核（Slave）（也可以称为备用核）。该从核（第二核）将第一核的标识（例如，编号等）写入该从核的主核（Master）编号寄存器中，第一核成为第二核的主核（Master），并将该从核（第二核）的标识（例如编号等）加入其备用核列表中。

例如，本发明实施例中可以在备用核列表中记录所有从核的标识。或者，本发明实施例中还可以用向量形式记录当前 Master 的所有 Slave，例如，如图 3 所示，多核处理器中共有 4 个核，那么备用核列表中记录的向量可以为 4 个比特位，第一到第四个比特位分别代表第一个核到第四个核，即 Core0-core3。例如，本发明实施例中的第一核可以是图 3 中的 Core3，表示该第一核的备用核列表的向量可以为 0100，其中，0 表示不是第一核的从核，1 表示为第一核的从核，那么根据备用核列表中的向量可知，4 个核中的第二个核为第一核的从核。

再例如，如图 4 所示，针对图 1 中的多核处理器为例，备用核列表可以为 16 个比特位的向量，其中从左至右的 16 个比特分别代表 Core0-Core15。以第一核为 core5 而言，第一核的备用核列表的向量例如可以为 0000001000000000；由于第 7 位为 1，因此，与第 7 位对应的核 Core6 为 Core5 的从核。

在实际应用中，主核获取到地址转换请求后，当主核本地的 TLB 表项缺失时，主核会读取从核中的 TLB 表项。下面描述主核读取从核中的 TLB 表项的具体过程。相应的作为另一实施例，在该第一核从该多核处理器处于空闲状态的核中确定第二核之后，该方法还可以包括：

该第一核接收第二地址转换请求，根据该第二地址转换请求查询该第一核内的 TLB；

在该第一核内的 TLB 中，该第一核确定和该第二地址转换请求对应的第二目标 TLB 表项缺失时，查询该第二核内的 TLB；

当在该第二核内的 TLB 中查询到该第二目标 TLB 表项时，将该第二目标 TLB 表项替换掉该第一核内的 TLB 中的第二表项，并将该第二表项存储

在该第二核内的 TLB 中。

例如，第一核将第二表项存储在第二核中原第二目标 TLB 表项的位置。也就是说，第一核将第二表项与第二目标 TLB 表项互换存储位置。

应理解，本实施例仅以第一核读取第二核的 TLB 中的第二目标表项为例进行说明，当第一核具有多个从核时，第一核可以向所有的从核发送读取第二目标表项的请求，第二目标表项也可以位于其他从核的 TLB 中，相应的读取与替换表项的过程与上述描述类似，这里不再赘述。

上述过程发生在 Master (第一核) 本地 TLB 缺失 (Miss)，即本地 TLB 不存在第二目标 TLB 表项的情况下，Master 会读取 Slave 中的 TLB 表项。例如，如图 5 所示。Master 根据备用核列表 (Slave List) (也称为从核列表)，向所有的 Slave 发送 TLB 读取请求。Slave 接收到 TLB 读取请求之后，查询从核本地的 TLB，如果第二目标 TLB Miss，则返回缺失反馈；如果命中，则返回命中反馈，以及命中 TLB 的表项内容。Master 收集所有反馈之后，若都是缺失反馈，则向操作系统发送 TLB 查询请求，进而从内存中获取该缺失的 TLB 表项；若有 Slave 命中，则利用命中 TLB 表项进行重填。当在重填过程若发生表项替换的话，则将被替换表项写到命中的从核中。

因此，本发明实施例中工作的核拥有空闲的核的 TLB 的读/写权限，通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，更进一步提高了第一核的 TLB 的容量，在第一核内的 TLB 缺失表项时，可以通过读取从核内的 TLB 获取目标表项，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

前面已经说明，在将被替换表项写入 Slave 的 TLB 的过程中，由 Slave 来保存写入位置。本发明实施例中可以采用轮流写入机制，即如果从核 (例如第二核) 的 TLB 表项数为 N，则从第一个表项开始，即从 0 开始依次写入直到 N-1。

当 N-1 位置 TLB 表项也被写入，即从核的 TLB 表项已存储满，表明从核的 TLB 中保存的都是 Master 的被替换表项，这里可以称此 Slave 为满从核 (Full Slave)；那么该 Slave 将无法在保持其他替换下来的表项，Full Slave 会向 Master 发送写溢出请求，Master 收到请求之后用 Full Slave 列表来记录该 Full Slave。

这时当 Master 替换下表项后,需要获取其他的从核用以存储该替换下来的表项。

即在 Master 中的有效 TLB 表项被替换,并且备用核列表中的所有 Slave 都是 Full Slave 的情况时,由于 Full Slave 的 TLB 中保存的都是 Master 的被替换表项,这些表项后续可能会被 Master 用到,因此,Master 希望获取更多的 Slave 来保存被替换表项。

相应地,作为另一实施例,该方法还可以包括:

该第一核接收第三地址转换请求,根据该第三地址转换请求查询该第一核内的 TLB;

10 在该第一核内的 TLB 中,该第一核确定和该第三地址转换请求对应的第三目标 TLB 表项缺失时,查询该第二核内的 TLB;

当在该第二核内的 TLB 中确定第三目标 TLB 表项缺失时,获取该第三目标 TLB 表项;

15 在判断该第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时,该第一核从该多核处理器处于空闲状态的核中确定第三核;

该第一核将该第三目标 TLB 表项替换掉该第一核内的 TLB 中的第三表项,并将该第三表项存储在该第三核内的 TLB 中。

因此,本发明实施例在当前的从核存储已全满后,会获取新的从核来保存替换下的 TLB 表项,进一步扩大了该第一核的容量,当该第一核再次查询该替换下来的 TLB 表项时,该第一核可以直接从该新的从核中读取,无需该核通过内存来获取。因此,本发明实施例能够加快程序的执行,提高程序的执行效率。

20 具体而言,在主核(第一核)的 TLB 中的表项已存储满,且该主核的所有从核(例如这里只有一个从核,为第二核)中的 TLB 表项已存储时,该主核(第一核)从该主核从多核处理器中的其他空闲核中确定第三核用于存储被第三目标 TLB 表项替换下来的第一核的 TLB 中的第三表项。

确定第三核的过程可以参照确定第二核的过程,此处不再赘述。

30 例如,如图 7 所示。在 Master 已有的 Slave 都是 Full Slave,如果继续将被替换表项写入 Full Slave 的 TLB 中,会覆盖之前的被替换表项。为了避免这种情况,Master 会重复上文中描述的 Slave 获取过程,获得图中新的 Slave 1(例如为第三核),并将被替换表项(例如为第三表项)写入 Slave 1 中。

上文中描述了主核确定从核以及使用从核的资源来存储 TLB 表项的方案。在当主核从工作状态转变到空闲状态后，由于不再使用 TLB 资源，因此，需要释放上述得到的从核资源。

相应地，作为另一实施例，当该第一核由工作状态转换到空闲状态后，

5 该方法还包括：

该第一核向该 TLB 备用核列表中记录的核发送 TLB 释放指令，该 TLB 释放指令用于指示该备用核列表中记录的核解除 TLB 共享。

也就是说，Master 变成空闲状态：释放所有已经获取的 TLB 资源。Master 根据 Slave List 向所有 Slave 发送 TLB 释放请求，这样这些空闲的核的 TLB 资源可以被其他工作的核使用。例如，如图 8 所示，Master 释放所有已经获取的 TLB 资源后，该主核和从核均成为空闲的核，可以被其他工作的核使用。

因此，本发明实施例在第一核转换到空闲状态后，通过向所有的从核发送释放指令释放已获取的从核的 TLB 资源，通过这种方式，使得这些从核（空闲的核）的 TLB 资源释放，避免资源的浪费，进而该释放的核的 TLB 资源能够被其他的工作的核使用，提升其他工作核的容量，加快工作的核的程序执行。

其中，从核解除 TLB 共享后，该从核可以将自身存储的 TLB 表项删除，也可以不删除，本发明实施例并不对此做限定。例如，在解出共享后，在该从核又作为另一个工作的核的从核后，可以将 TLB 中的表项全部删除。以供该另一个工作的核存储替换下的表项。在例如，在解除共享后，在该从核成为工作状态时，可以保留之前存储的 TLB 表项，以供自身查找使用。

相类似地，当 Slave（例如，第二核）变成运行状态：Slave 会向 Master 发送 TLB 解除共享请求，Master 接收到请求之后将其从 Slave List 中去除。例如，如图 9 所示，当 Slave1 变成运行状态后，主核会释放该 Slave1，并从从核列表中删除。

相应地，作为另一实施例，该第二核由空闲状态转换到工作状态后，该方法还包括：

该第一核接收该第二核发送的 TLB 解除共享请求，该 TLB 解除共享请求中携带有该第二核的标识；

在该 TLB 备用核列表中，该第一核将该第二核的标识删除。

因此，本发明实施例在第二核由空闲状态变为工作状态后，该第二核会向第一核发送 TLB 解除共享请求，进而使得第一核释放该第二核，使得第二核能够使用自身的 TLB 资源，避免了影响第二核正在的业务处理，同时，第二核还可以成为另一个主核，可以使用其他的空闲核的 TLB 资源。

5 进一步地，作为另一实施例，在该 TLB 备用核列表中，该第一核将该第二核的标识删除之前，该方法还包括：

该第一核从该多核处理器处于空闲状态的核中确定第四核；

该第一核将该第二核的 TLB 中的所有表项拷贝到该第四核的 TLB 中。

也就是说，在主核（例如，第一核）从备用核列表中删除从核（例如，10 第二核）之前，会从多核处理器中重新获取一个核（新的从核），该新的从核的 TLB 用于存储删除的从核（即第二核）的 TLB 中所有的表项。这样在后续查询中如果需要查询第二核中的表项，不需要通过操作系统来重新获取这些表项，通过直接访问从核（第四核）的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

15 确定第四核的过程可以参照确定第二核的过程，此处不再赘述。

例如，如图 10 所示，当 Slave1 变成运行状态后，主核会释放该 Slave1，并从从核列表中删除。同时，主核会重新确定一个从核，例如 Slave 2，将 Slave 1 的 TLB 中的所有表项存储在 Slave 2 的 TLB 中，并在从核列表中记录该 Slave 2。

20 因此，本发明实施例通过第一核获取新的从核（第四核），来保存释放的第二核内的 TLB 中的所有表项，在该第一核在后续查询中如果需要再次查询第二核中的表项，不需要通过操作系统从内存中重新获取这些表项，而可以直接通过访问从核（第四核）的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

25 需要说明的是，本发明实施例中的该多核处理器中的每一个核均设置有标志位寄存器，该标志位寄存器用于记录该核的状态标志位、主核（Master）标志位和从核（Slave）标志位；

该状态标志位用于表示该核的运行状态和共享状态，该运行状态包括空闲状态或工作状态，该共享状态包括主核状态、从核状态或不参与共享状态，30 该主核状态表示该核处于工作状态且使用其他空闲核的 TLB 资源管理 TLB 表项，该从核状态表示该核处于空闲状态且为主核共享 TLB 资源；

该主核标志位用于表示在该核为主核时的空闲核列表、从核 (Slave) 列表 (也称为备用核列表) 和全满的从核 (Full Slave) 列表, 该空闲核列表包括用于表示所有空闲核的向量, 该从核列表包括用于表示该核的所有从核的向量, 该全满的从核列表包括用于表示该核的所有满从核的向量;

- 5 该从核标志位用于表示在该核为从核时的主核 (Master) 编号和替换表项的写入位置, 该主核编号包括该核的唯一主核的标识, 该替换表项的写入位置包括从主核中被替换表项在从核中的写入位置。

10 具体而言, 为了实现本发明提出的 TLB 资源共享方法, 需要在每个节点中增加一些寄存器来保存标志位, 包括状态标志位, Master 标志位和 Slave 标志位。由于每个节点都可能成为 Master 或 Slave, 所以每个节点中都配备了这三种标志位。例如如表 1 所示。

表1 每个节点中的标志位的描述

类型	名称	位宽	描述
状态标志位	运行状态位	1	0: 空闲节点; 1: 工作节点
	共享状态位	2	00: 不参与共享; 01: Master; 10: Slave
master标志位	空闲节点列表	节点数	向量形式记录系统中的空闲节点, 0 表示运行, 1 表示空闲
	Slave 列表	节点数	向量形式记录当前 Master 的所有 Slave, 0 表示非 Slave, 1 表示 Slave
	full Slave 列表	节点数	向量形式记录当前 Master 的所有 full Slave, 0 表示非 Full Slave, 1 表示 full Slave
Slave标志位	Master 编号	$\lceil \log_2(\text{节点数}) \rceil$	记录 Slave 对应的唯一 Master 编号
	写入位置	$\lceil \log_2(\text{TLB 表项数}) \rceil$	记录 Master 被替换表项在 Slave TLB 中的写入位置

15 下面对各个标志位进行详细的阐述:

运行状态位：区分空闲/工作节点，所以可以只需要 1 bit 位宽的寄存器。

共享状态位：区分 Master/Slave/不参与共享节点，所以需要 2 bits 位宽的寄存器。

空闲节点/Slave/Full Slave 列表：功能在上文中已经有过详细描述。在寄存器实现中采用向量的方式，向量的宽度等于系统中节点的数量，向量中每一比特（bit）对应一个节点。以 Slave 列表为例，向量中某一比特为 0 表示对应节点不是 Slave，为 1 表示对应节点是 Slave。

Master 编号：Slave 只能有唯一的 Master，对 Slave 而言，Master 编号中记录了其 Master 的编号；当 Slave 变成运行状态时，可以根据 Master 编号来通知对应的 Master。Master 编号的二进制位宽可以为 $\lceil \log_2(\text{节点数}) \rceil$ ，即为大于或等于 $\log_2(\text{节点数})$ 的最小整数，例如，当多核处理器包括 8 个核（8 个节点）时，表示 Master 编号需要 $\lceil \log_2(8) \rceil$ 即 3 个比特。在例如，当多核处理器包括 12 个核（12 个节点）时，表示 Master 编号需要 $\lceil \log_2(12) \rceil$ 即 4 个比特。

写入位置：由 Slave 来进行维护，表示接收到 Master 被替换表项时的写入位置。其二进制位宽可以为 $\lceil \log_2(\text{TLB 表项数}) \rceil$ 即大于或等于 $\log_2(\text{TLB 表项数})$ 的最小整数。例如，当 Slave 内的 TBL 的表项为 64 个，那么表示写入位置需要 $\lceil \log_2(64) \rceil$ 即 6 个比特。

因此，本发明实施例通过标志位的设置，实现了主核对从核内的 TLB 资源进行读写，进而扩大了主核的 TLB 的容量，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

上文中结合图 1 至图 10 详细描述本发明实施例的管理 TLB 的方法。下面将结合图 11 详细描述本发明实施例的多核处理器。

图 11 是根据本发明一个实施例的多核处理器 1100 的示意框图。

如图 11 所示，该多核处理器 1100 包括第一核 1110 和第二核 1120。可选地，还可以包括第三核 1130，可选的，还可以包括第四核 1140。如图 1 类似，该多核处理器 1100 中的每一个核都包括处理模块；缓存模块，例如缓存模块包括一级缓存（L1）和二级缓存（L2）；和片上网络接口（Switch）。每个核中的处理模块内包含有一个 TLB。其中，各个核之间通过片上网络连接，通过片上网络接口互相通信。

应理解，本发明实施例中的多核处理器 1100 还可以包括更多的核，例如，多核处理器 1100 可以包括 8 个核、10 核、16 个核、32 个核等，本发明实施例并不对此做限定。

还应理解，本发明实施例中，第一核可以是多核处理器中的任意一个核，
5 为了表示方便，在图 1 中以第一个核作为第一核进行描述，但本发明实施例并不限于此。还应理解，在实际应用中，第二核也可以是多核处理器中除第一核之外的任意一个核，也就是说第二核可能不直接与第一核相连，这里为表示的方便，在图 11 中以第二核和第一核直接相连为例进行描述，但本发明实施例并不限于此。类似地，下文中的第三核和第四核在图 11 中为了展
10 示的方便，直接与第一核或第二核直接相连，但本发明实施例并不限于此。

应理解图 11 所示的多核处理器与图 1 至图 10 方法实施例相对应，图 11 中的多核处理器 1100 能够实现图 1 至图 10 涉及的方法的各个过程，为了避免重复，本发明实施例适当省略详细描述。

具体而言，该第一核 1110 用于接收第一地址转换请求，根据该第一地址转换请求查询该第一核内的 TLB；
15

在该第一核内的 TLB 中，确定和该第一地址转换请求对应的第一目标 TLB 表项缺失时，获取该第一目标 TLB 表项；

在判断该第一核内的 TLB 中的表项存储已满时，从该多核处理器中处于空闲状态的核中确定第二核；将该第一目标 TLB 表项替换掉该第一核内的 TLB 中的第一表项。
20

该第二核 820 用于将该第一表项存储在该第二核内的 TLB 中。

应理解，本发明实施例中的第一核可以称为第二核的主核，第二核可以称为第一核的从核。

因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项后替换第一核的 TLB 中的第一表项，并将第一表项存储在第二核的 TLB 中，由于将替换下来的表项存储在了第二核的 TLB 中。因此，本发明实施例通过利用空闲核的 TLB 资源来扩大工作的核的 TLB 容量，从而能够降低 TLB 缺失率，加快程序的执行。
25

由于工作的核即主核 (Master) (例如，第一核) 将被替换的 TLB 表项写入空闲节点即从核 (Slave) (例如，第二核) 的 TLB 中。当工作的核需要再次用到这些被替换的 TLB 表项时，不需要通过操作系统来重新获取这些
30

表项，通过直接访问 Slave 的 TLB 就可以获取这些表项，因此大大降低了 TLB 重填的时延，加快了程序的执行。

应注意，本发明实施例中一个空闲的核仅能将自身的 TLB 资源共享给一个工作的核使用；一个工作的核可以获取多核空闲的核的 TLB 资源存储

5 TLB 表项。

可选地，在获取该第一目标 TLB 表项时，该第一核 1110 具体用于：

从内存的页表中获取该第一目标 TLB 表项，

或者，从该多核系统中的其他核获取该第一目标 TLB 表项。

例如，第一核可以向多核系统中的其他核广播 TLB 查询请求，并在该
10 广播的 TLB 查询请求中携带导致缺失的虚拟地址，即第一目标 TLB 表项对应的虚拟地址，其他的核收到广播地址后，在本地的 TLB 中查找此虚拟地址，如果其中一个处理器核内的 TLB 命中，则可以将该第一目标 TLB 表项反馈给该第一核。

通过这种方式，使得第一核可以从其他的核中快速的获的该第一目标
15 TLB 表项，避免该第一核向操作系统发起查询请求，从内存中获取该第一目标 TLB 表项，能够节省一定的时间，提高了应用效率。

另外，第一核也可以是从内存的页表中获得该第一目标 TLB 表项，例如，第一核向操作系统发送查询请求，查询请求中携带导致缺失的虚拟地址，经过操作系统处理从内存的页表中获得该第一目标 TLB 表项。本发明实施
20 例并不限于此。

进一步地，在从该多核处理器中处于空闲状态的核中确定第二核时，该第一核 1110 具体用于：

向该多核处理器中其他核中每个核发送状态查询请求，该状态查询请求用于查询该每个核是否处于空闲状态；

25 接收该每个核发送的响应消息，该响应消息用于指示该每个核是否处于空闲状态；

并根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核。

因此，本发明实施例通过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，还间接的提高了第一核
30 的 TLB 的容量，减低了第一核通过内存来获取目标 TLB 表项的发生，加快程序的执行。

进一步地，在根据该响应消息，从处于空闲状态的核中选择一个核作为该第二核时，该第一核 1110 具体用于：

根据该响应消息确定空闲核列表，该空闲核列表中包括该多核处理器中除该第一核外的其他核中处于空闲状态的核；

5 在该空闲核列表中，选择与该第一核的通信开销最小的空闲核作为该第二核。

因此，本发明实施例中第一核选择与该第一核通信开销最小的空闲核作为第二核，并将替换下的 TLB 表项存储在第二核中，最大限度的降低通信开销。并且在需要查询第二核内的 TLB 的表项时，由于通信开销很小，所以第一核能够快速查询到该 TLB 表项，提高程序的执行效率。

进一步地，在选择与该第一核的通信开销最小的空闲核作为该第二核时，该第一核 1110 具体用于：

将该空闲核列表中与该第一核的通信跳数最小的空闲核作为该第二核；或者，

15 将该空闲核列表中与该第一核的物理距离最小的空闲核作为该第二核。

可选地，在确定该第二核之后，该第一核 1110 还用于将该第二核的标识记录在该第一核内的 TLB 备用核列表中。

因此，本发明实施例，在第一核的 TLB 表项存储已满且第一目标 TLB 表项缺失，在获取到该第一目标 TLB 表项确定第二核，并将第二核记录在该第一核的备用核列表中，这样第一核根据备用核列表可以对备用核列表中的第二核内的 TLB 进行读写，进而提升第一核的 TLB 的容量，从而能够降低第一核的 TLB 缺失率，加快程序的执行。

可选地，该第一核 1110 还用于：

25 接收第二地址转换请求，根据该第二地址转换请求查询该第一核内的 TLB；

在该第一核内的 TLB 中，确定和该第二地址转换请求对应的第二目标 TLB 表项缺失时，查询该第二核内的 TLB；

当在该第二核内的 TLB 中查询到该第二目录 TLB 表项时，将该第二目标 TLB 表项替换掉该第一核内的 TLB 中的第二表项，

30 该第二核 1120 还用于将该第二表项存储在该第二核内的 TLB 中。

因此，本发明实施例中工作的核拥有空闲的核的 TLB 的读/写权限，通

过使用空闲核内的 LTB 资源来保存被替换下的 TLB 表项，不但提高了空闲核 TLB 资源的利用率，更进一步提高了第一核的 TLB 的容量，在第一核内的 TLB 缺失表项时，可以通过读取从核内的 TLB 获取目标表项，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

5 可选地，该第一核 1110 还用于：

接收第三地址转换请求，根据该第三地址转换请求查询该第一核内的 TLB；

10 在该第一核内的 TLB 中，确定和该第三地址转换请求对应的第三目标 TLB 表项缺失时，查询该第二核内的 TLB；

当在该第二核内的 TLB 中确定第三目标 TLB 表项缺失时，获取该第三目标 TLB 表项；

在判断该第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时，从该多核处理器处于空闲状态的核中确定第三核 1130；

15 将该第三目标 TLB 表项替换掉该第一核内的 TLB 中的第三表项，
该第三核用于将该第三表项存储在该第三核内的 TLB 中。

因此，本发明实施例在当前的从核存储已全满后，会获取新的从核来保存替换下的 TLB 表项，进一步扩大了该第一核的容量，当该第一核再次查询该替换下来的 TLB 表项时，该第一核可以直接从该新的从核中读取，无需该核通过内存来获取。因此，本发明实施例能够加快程序的执行，提高程序的执行效率。

20 可选地，当该第一核由工作状态转换到空闲状态后，该第一核还用于：

向该 TLB 备用核列表中记录的核发送 TLB 释放指令，该 TLB 释放指令用于指示该备用核列表中记录的核解除 TLB 共享。

25 因此，本发明实施例在第一核转换到空闲状态后，通过向所有的从核发送释放指令释放已获取的从核的 TLB 资源，通过这种方式，使得这些从核（空闲的核）的 TLB 资源释放，避免资源的浪费，进而该释放的核的 TLB 资源能够被其他的工作的核使用，提升其他工作核的容量，加快工作的核的程序执行。

30 可选地，当该第二核由空闲状态转换到工作状态后，
所第二核用于向该第一核发送的 TLB 解除共享请求，该 TLB 解除共享

请求中携带有该第二核的标识;

该第一核还用于接收该 TLB 解除共享请求, 在该 TLB 备用核列表中, 将该第二核的标识删除。

因此, 本发明实施例在第二核由空闲状态变为工作状态后, 该第二核会
5 向第一核发送 TLB 解除共享请求, 进而使得第一核释放该第二核, 使得第二核能够使用自身的 TLB 资源, 避免了影响第二核正在的业务处理, 同时, 第二核还可以成为另一个主核, 可以使用其他的空闲核的 TLB 资源。

可选地, 该第一核还用于从该多核处理器处于空闲状态的核中确定第四核 1140;

10 该第四核的 TLB 用于存储该第二核的 TLB 中的所有表项。

因此, 本发明实施例通过第一核获取新的从核 (第四核), 来保存释放的第二核内的 TLB 中的所有表项, 在该第一核在后续查询中如果再次需要查询第二核中的表项, 不需要通过操作系统从内存中重新获取这些表项, 而可以直接通过访问从核 (第四核) 的 TLB 就可以获取这些表项, 因此大大
15 降低了 TLB 重填的时延, 加快了程序的执行。

需要说明的是, 本发明实施例中的该多核处理器中的每一个核均设置有标志位寄存器, 该标志位寄存器用于记录该核的状态标志位、主核 (Master) 标志位和从核 (Slave) 标志位;

该状态标志位用于表示该核的运行状态和共享状态, 该运行状态包括空闲状态或工作状态, 该共享状态包括主核状态、从核状态或不参与共享状态, 该主核状态表示该核处于工作状态且使用其他空闲核的 TLB 资源管理 TLB 表项, 该从核状态表示该核处于空闲状态且为主核共享 TLB 资源;
20

该主核标志位用于表示在该核为主核时的空闲核列表、从核 (Slave) 列表 (也称为备用核列表) 和全满的从核 (Full Slave) 列表, 该空闲核列表包括用于表示所有空闲核的向量, 该从核列表包括用于表示该核的所有从核的向量, 该全满的从核列表包括用于表示该核的所有满从核的向量;
25

该从核标志位用于表示在该核为从核时的主核 (Master) 编号和替换表项的写入位置, 该主核编号包括该核的唯一主核的标识, 该替换表项的写入位置包括从主核中被替换表项在从核中的写入位置。

30 具体而言, 为了实现本发明提出的 TLB 资源共享方法, 需要在每个节点中增加一些寄存器来保存标志位, 包括状态标志位, Master 标志位和 Slave

标志位。由于每个节点都可能成为 Master 或 Slave，所以每个节点中都配备了这三种标志位。例如，如上文表 1 所示，此处不再赘述。

因此，本发明实施例通过标志位的设置，实现了主核对从核内的 TLB 资源进行读写，进而扩大了主核的 TLB 的容量，也即减低了第一核通过内存来获取目标 TLB 表项的发生，能够加快程序的执行，提高程序的执行效率。

应理解，说明书中提到的“一个实施例”或“一实施例”意味着与实施例有关的特定特征、结构或特性包括在本发明的至少一个实施例中。因此，在整个说明书各处出现的“在一个实施例中”或“在一实施例中”未必一定指相同的实施例。此外，这些特定的特征、结构或特性可以任意适合的方式结合在一个或多个实施例中。应理解，在本发明的各种实施例中，上述各过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不对本发明实施例的实施过程构成任何限定。

另外，本文中术语“系统”和“网络”在本文中常被可互换使用。本文中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

应理解，在本发明实施例中，“与 A 相应的 B”表示 B 与 A 相关联，根据 A 可以确定 B。但还应理解，根据 A 确定 B 并不意味着仅仅根据 A 确定 B，还可以根据 A 和/或其它信息确定 B。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另外，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、装置或单元的间接耦合或通信连接，也可以是电的，机械的或其它的形式连接。

作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本发明实施例方案的目的。

另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以是两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

通过以上的实施方式的描述，所属领域的技术人员可以清楚地了解到本发明可以用硬件实现，或固件实现，或它们的组合方式来实现。当使用软件实现时，可以将上述功能存储在计算机可读介质中或作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质，其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是计算机能够存取的任何可用介质。以此为例但不限于：计算机可读介质可以包括RAM、ROM、EEPROM、CD-ROM或其他光盘存储、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质。此外，任何连接可以适当的成为计算机可读介质。例如，如果软件是使用同轴电缆、光纤光缆、双绞线、数字用户线（DSL）或者诸如红外线、无线电和微波之类的无线技术从网站、服务器或者其他远程源传输的，那么同轴电缆、光纤光缆、双绞线、DSL或者诸如红外线、无线和微波之类的无线技术包括在所属介质的定义中。如本发明所使用的，盘（Disk）和碟（disc）包括压缩光碟（CD）、激光碟、光碟、数字通用光碟（DVD）、软盘和蓝光光碟，其中盘通常磁性的复制数据，而碟则用激光来光学的复制数据。上面

的组合也应当包括在计算机可读介质的保护范围之内。

总之，以上所述仅为本发明技术方案的较佳实施例而已，并非用于限定本发明的保护范围。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

权利要求

1. 一种管理转址旁路缓存 TLB 的方法，其特征在于，应用于多核处理器，所述多核处理器包括第一核，所述第一核内包含一个 TLB，所述方法包括：

5 所述第一核接收第一地址转换请求，根据所述第一地址转换请求查询所述第一核内的 TLB；

在所述第一核内的 TLB 中，所述第一核确定和所述第一地址转换请求对应的第一目标 TLB 表项缺失时，获取所述第一目标 TLB 表项；

10 在判断所述第一核内的 TLB 中的表项存储已满时，所述第一核从所述多核处理器中处于空闲状态的核中确定第二核；

所述第一核将所述第一目标 TLB 表项替换掉所述第一核内的 TLB 中的第一表项，并将所述第一表项存储在所述第二核内的 TLB 中。

2. 根据权利要求 1 所述的方法，其特征在于，所述第一核从所述多核处
15 理器中处于空闲状态的核中确定第二核，包括：

所述第一核向所述多核处理器中其他核中每个核发送状态查询请求，所述状态查询请求用于查询所述每个核是否处于空闲状态；

所述第一核接收所述其他核中每个核发送的响应消息，所述响应消息用于指示所述每个核是否处于空闲状态；

20 所述第一核根据所述响应消息，从处于空闲状态的核中选择一个核作为所述第二核。

3. 根据权利要求 2 所述的方法，其特征在于，所述第一核根据所述响应消息，从处于空闲状态的核中选择一个核作为所述第二核，包括：

25 所述第一核根据所述响应消息确定空闲核列表，所述空闲核列表中包括所述多核处理器中除所述第一核外的其他核中处于空闲状态的核；

在所述空闲核列表中，选择与所述第一核的通信开销最小的空闲核作为所述第二核。

30 4. 根据权利要求 3 所述的方法，其特征在于，在所述空闲核列表中，选择与所述第一核的通信开销最小的空闲核作为所述第二核，包括：

所述第一核将所述空闲核列表中与所述第一核的通信跳数最小的空闲核作为所述第二核；或者，

所述第一核将所述空闲核列表中与所述第一核的物理距离最小的空闲核作为所述第二核。

5

5.根据权利要求 1 至 4 中任一所述的方法，其特征在于，在确定所述第二核之后，所述方法还包括：

将所述第二核的标识记录在所述第一核内的 TLB 备用核列表中。

10

6. 根据权利要求 1 至 5 中任一项所述的方法，其特征在于，在所述第一核从所述多核处理器处于空闲状态的核中确定第二核之后，所述方法还包括：

所述第一核接收第二地址转换请求，根据所述第二地址转换请求查询所述第一核内的 TLB；

15

在所述第一核内的 TLB 中，所述第一核确定和所述第二地址转换请求对应的第二目标 TLB 表项缺失时，查询所述第二核内的 TLB；

当在所述第二核内的 TLB 中查询到所述第二目录 TLB 表项时，将所述第二目标 TLB 表项替换掉所述第一核内的 TLB 中的第二表项，并将所述第二表项存储在所述第二核内的 TLB 中。

20

7. 根据权利要求 1 至 5 中任一项所述的方法，其特征在于，所述方法还包括：

所述第一核接收第三地址转换请求，根据所述第三地址转换请求查询所述第一核内的 TLB；

25

在所述第一核内的 TLB 中，所述第一核确定和所述第三地址转换请求对应的第三目标 TLB 表项缺失时，查询所述第二核内的 TLB；

当在所述第二核内的 TLB 中确定第三目标 TLB 表项缺失时，获取所述第三目标 TLB 表项；

30

在判断所述第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时，所述第一核从所述多核处理器处于空闲状态的核中确定第三核；

所述第一核将所述第三目标 TLB 表项替换掉所述第一核内的 TLB 中的第三表项，并将所述第三表项存储在所述第三核内的 TLB 中。

8. 根据权利要求 5 至 7 中任一项所述的方法，其特征在于，当所述第一核由工作状态转换到空闲状态后，所述方法还包括：

所述第一核向所述 TLB 备用核列表中记录的核发送 TLB 释放指令，所述 TLB 释放指令用于指示所述备用核列表中记录的核解除 TLB 共享。

9. 根据权利要求 5 至 7 中任一项所述的方法，其特征在于，当所述第二核由空闲状态转换到工作状态后，所述方法还包括：

所述第一核接收所述第二核发送的 TLB 解除共享请求，所述 TLB 解除共享请求中携带有所述第二核的标识；

在所述 TLB 备用核列表中，所述第一核将所述第二核的标识删除。

10. 根据权利要求 9 所述的方法，其特征在于，在所述 TLB 备用核列表中，所述第一核将所述第二核的标识删除之前，所述方法还包括：

所述第一核从所述多核处理器处于空闲状态的核中确定第四核；

所述第一核将所述第二核内的 TLB 中的所有表项拷贝到所述第四核内的 TLB 中。

20

11. 根据权利要求 1 至 10 中任一项所述的方法，其特征在于，所述获取所述第一目标 TLB 表项，包括：

所述第一核从内存的页表中获取所述第一目标 TLB 表项，

或者，所述第一核从所述多核系统中的其他核获取所述第一目标 TLB 表项。

25

12. 一种多核处理器，其特征在于，所述多核处理器包括第一核，所述第一核内包括一个转址旁路缓存 TLB，

所述第一核用于接收第一地址转换请求，根据所述第一地址转换请求查询所述第一核内的 TLB，在所述第一核内的 TLB 中，确定和所述第一地址转换请求对应的第一目标 TLB 表项缺失时，获取所述第一目标 TLB 表项，

30

在判断所述第一核内的 TLB 中的表项存储已满时，从所述多核处理器中处于空闲状态的核中确定第二核，将所述第一目标 TLB 表项替换掉所述第一核内的 TLB 中的第一表项；

所述第二核用于将所述第一表项存储在所述第二核内的 TLB 中。

5

13. 根据权利要求 12 所述的多核处理器，其特征在于，在从所述多核处理器中处于空闲状态的核中确定第二核时，所述第一核具体用于：

10 向所述多核处理器中其他核中每个核发送状态查询请求，所述状态查询请求用于查询所述每个核是否处于空闲状态；接收所述每个核发送的响应消息，所述响应消息用于指示所述每个核是否处于空闲状态；并根据所述响应消息，从处于空闲状态的核中选择一个核作为所述第二核。

14. 根据权利要求 13 所述的多核处理器，其特征在于，在根据所述响应消息，从处于空闲状态的核中选择一个核作为所述第二核时，所述第一核具体用于：

15 根据所述响应消息确定空闲核列表，所述空闲核列表中包括所述多核处理器中除所述第一核外的其他核中处于空闲状态的核；在所述空闲核列表中，选择与所述第一核的通信开销最小的空闲核作为所述第二核。

20 15. 根据权利要求 14 所述的多核处理器，其特征在于，在选择与所述第一核的通信开销最小的空闲核作为所述第二核时，所述第一核具体用于：

将所述空闲核列表中与所述第一核的通信跳数最小的空闲核作为所述第二核；或者，

25 将所述空闲核列表中与所述第一核的物理距离最小的空闲核作为所述第二核。

16. 根据权利要求 12 至 15 中任一所述的多核处理器，其特征在于，在确定所述第二核之后，所述第一核还用于将所述第二核的标识记录在所述第一核内的 TLB 备用核列表中。

30

17. 根据权利要求 16 中任一项所述的多核处理器，其特征在于，所述

第一核还用于:

接收第二地址转换请求, 根据所述第二地址转换请求查询所述第一核内的 TLB;

5 在所述第一核内的 TLB 中, 确定和所述第二地址转换请求对应的第二目标 TLB 表项缺失时, 查询所述第二核内的 TLB;

当在所述第二核内的 TLB 中查询到所述第二目录 TLB 表项时, 将所述第二目标 TLB 表项替换掉所述第一核内的 TLB 中的第二表项,

所述第二核还用于将所述第二表项存储在所述第二核内的 TLB 中。

10 18. 根据权利要求 16 中所述的多核处理器, 其特征在于, 所述第一核还用于:

接收第三地址转换请求, 根据所述第三地址转换请求查询所述第一核内的 TLB;

15 在所述第一核内的 TLB 中, 确定和所述第三地址转换请求对应的第三目标 TLB 表项缺失时, 查询所述第二核内的 TLB;

当在所述第二核内的 TLB 中确定第三目标 TLB 表项缺失时, 获取所述第三目标 TLB 表项;

在判断所述第一核内的 TLB 中的表项以及第二核内的 TLB 中的表项存储都已满时, 从所述多核处理器处于空闲状态的核中确定第三核;

20 将所述第三目标 TLB 表项替换掉所述第一核内的 TLB 中的第三表项, 所述第三核用于将所述第三表项存储在所述第三核内的 TLB 中。

19. 根据权利要求 16 至 18 中任一项所述的多核处理器, 其特征在于, 当所述第一核由工作状态转换到空闲状态后, 所述第一核还用于:

25 向所述 TLB 备用核列表中记录的核发送 TLB 释放指令, 所述 TLB 释放指令用于指示所述备用核列表中记录的核解除 TLB 共享。

20. 根据权利要求 16 至 18 中任一项所述的多核处理器, 其特征在于, 当所述第二核由空闲状态转换到工作状态后,

30 所述第二核还用于向所述第一核发送的 TLB 解除共享请求, 所述 TLB 解除共享请求中携带有所述第二核的标识;

所述第一核还用于接收所述 TLB 解除共享请求，在所述 TLB 备用核列表中，将所述第二核的标识删除。

21. 根据权利要求 20 所述的多核处理器，其特征在于，当所述第二核
5 由空闲状态转换到工作状态后，

所述第一核还用于从所述多核处理器处于空闲状态的核中确定第四核，
并将所述第二核内的 TLB 中的所有表项拷贝到所述第四核内的 TLB 中。

22. 根据权利要求 12 至 21 中任一项所述的多核处理器，其特征在于，
10 在获取所述第一目标 TLB 表项时，所述第一核具体用于：

从内存的页表中获取所述第一目标 TLB 表项，

或者，从所述多核系统中的其他核获取所述第一目标 TLB 表项。

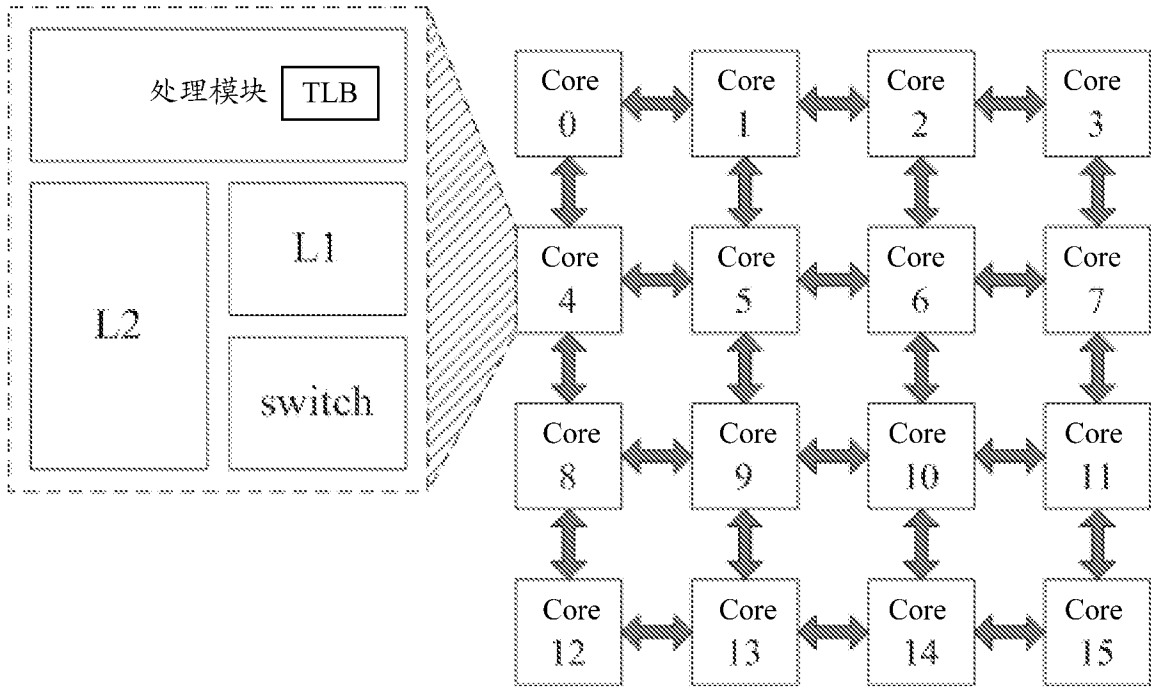


图 1

200

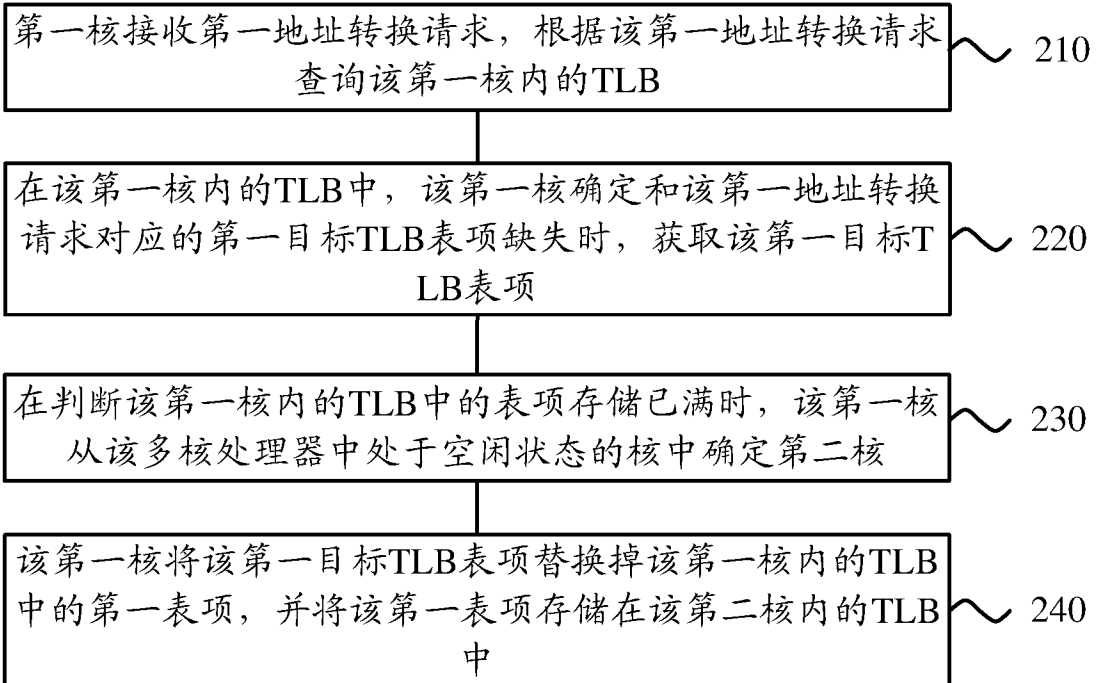


图 2

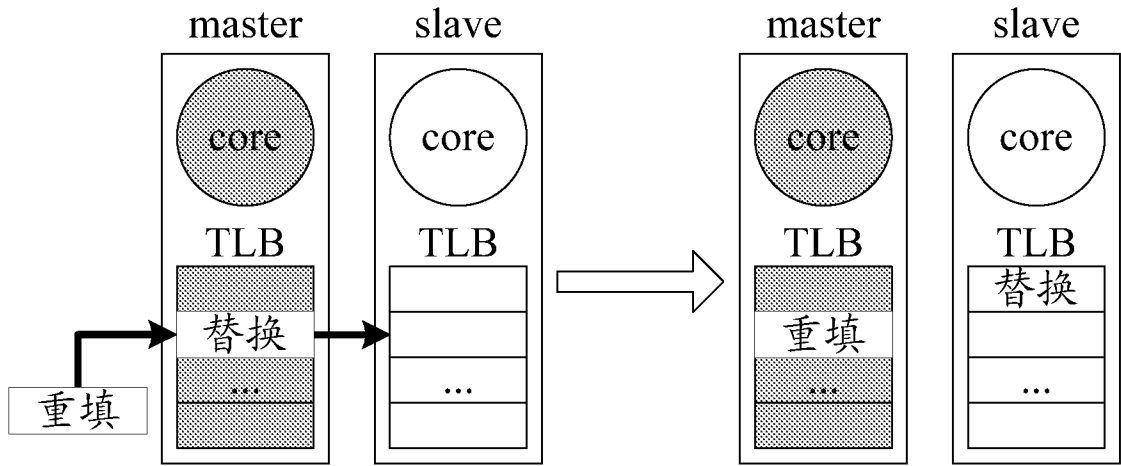


图 3

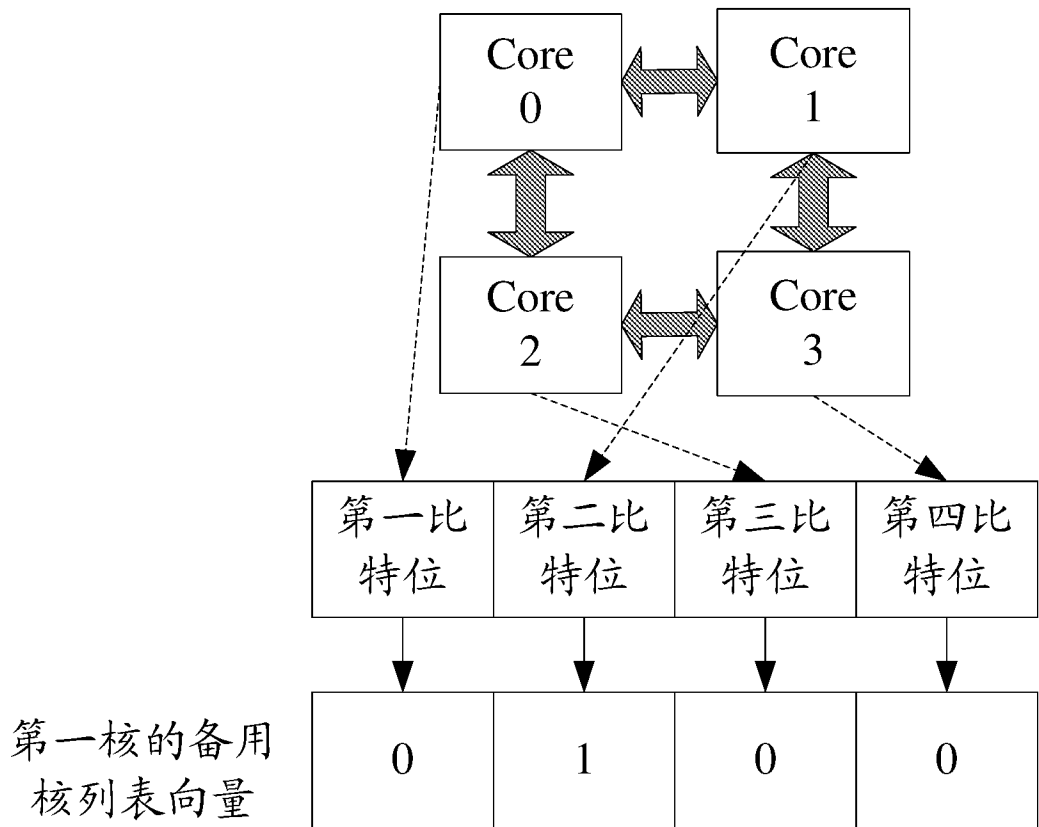


图 4

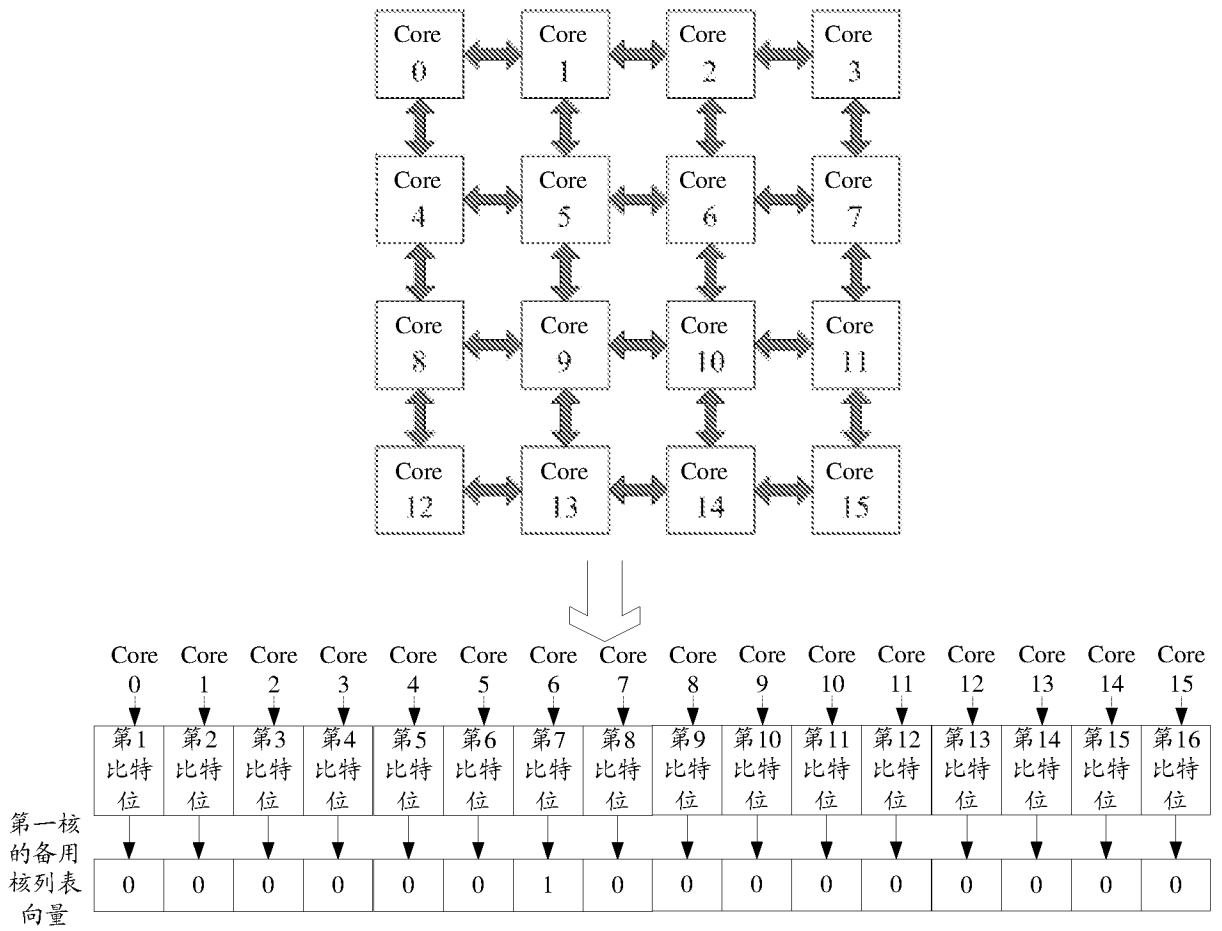


图 5

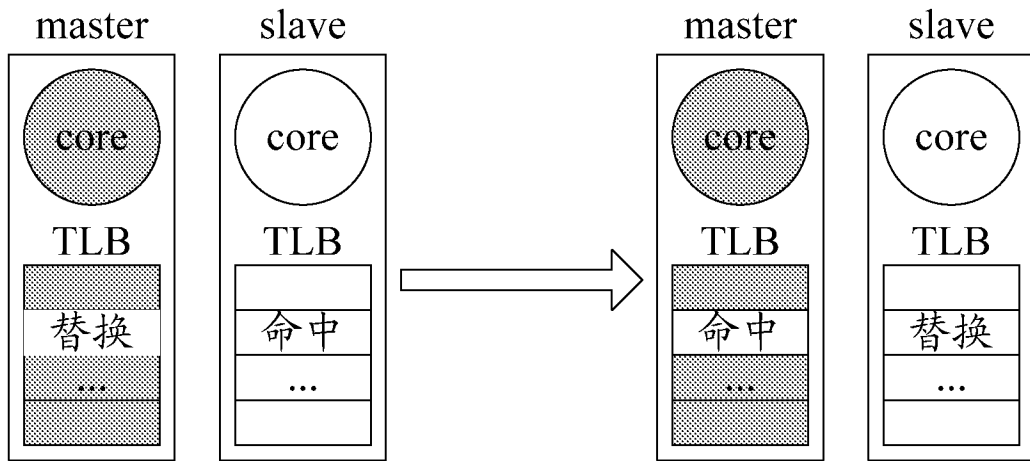


图 6

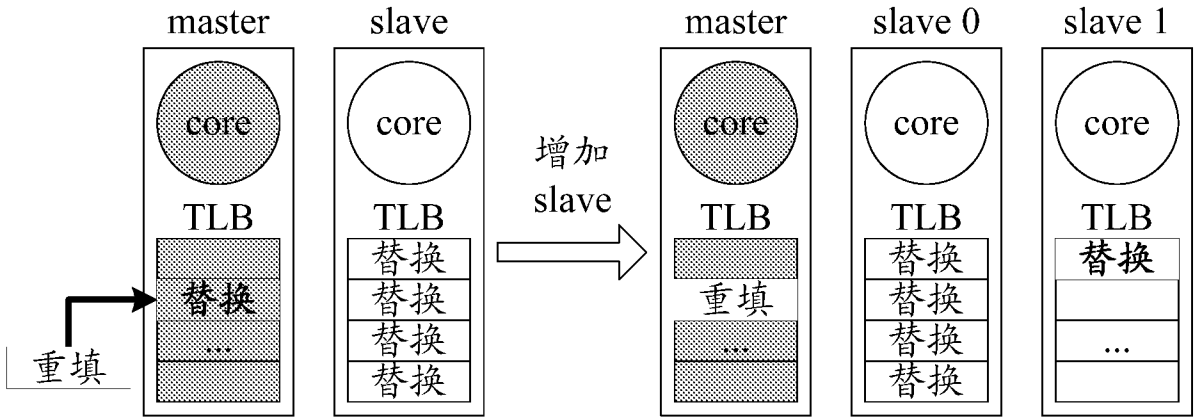


图 7

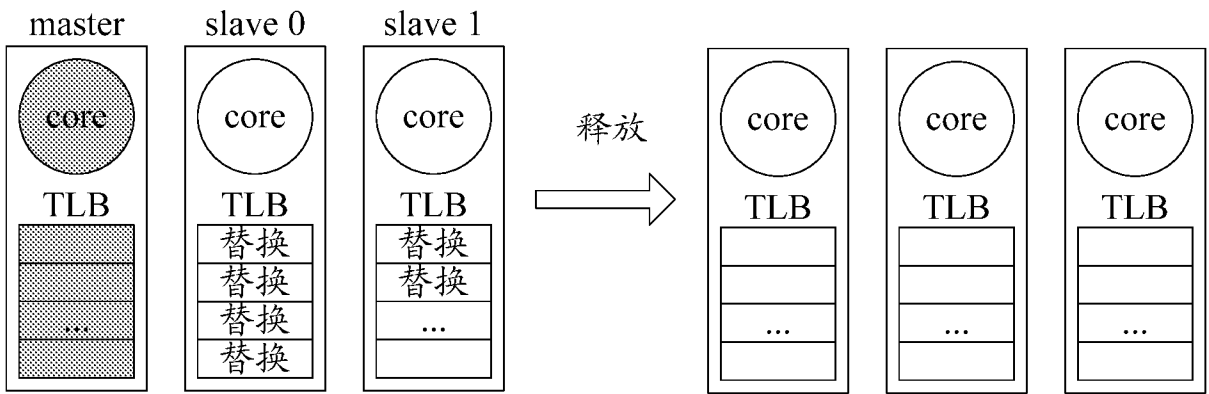


图 8

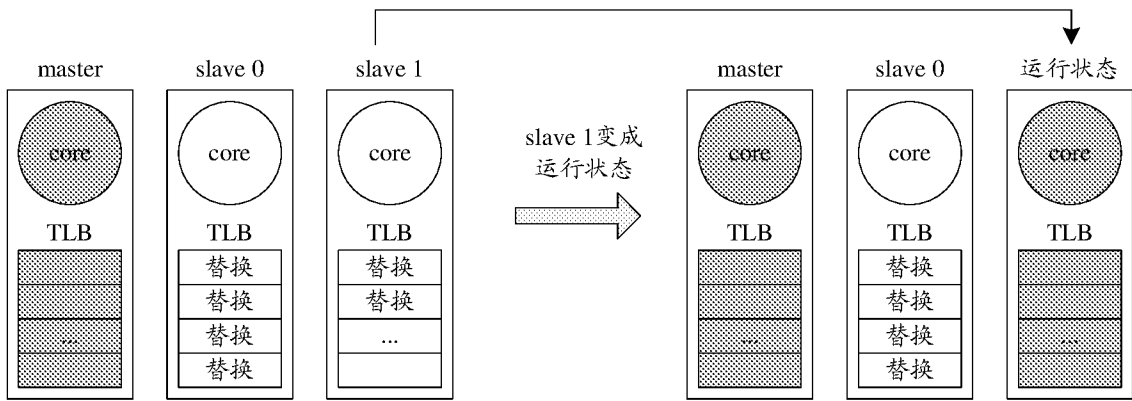


图 9

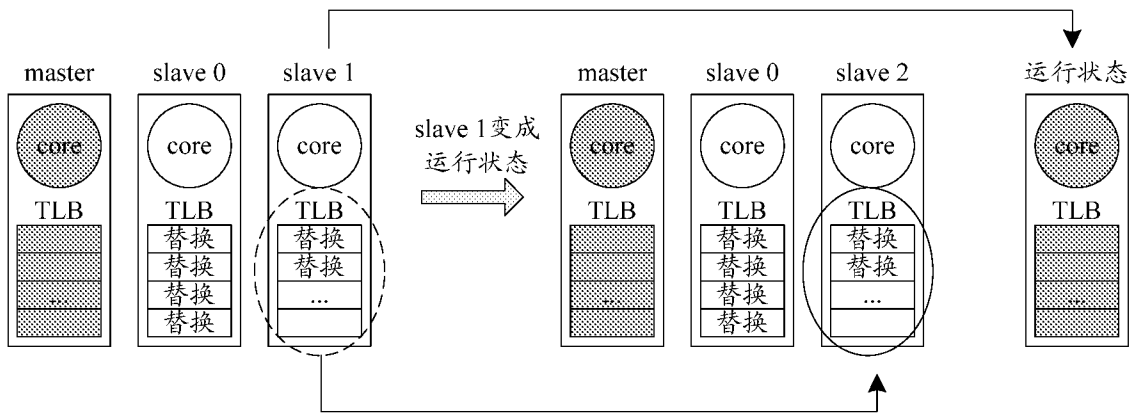


图 10

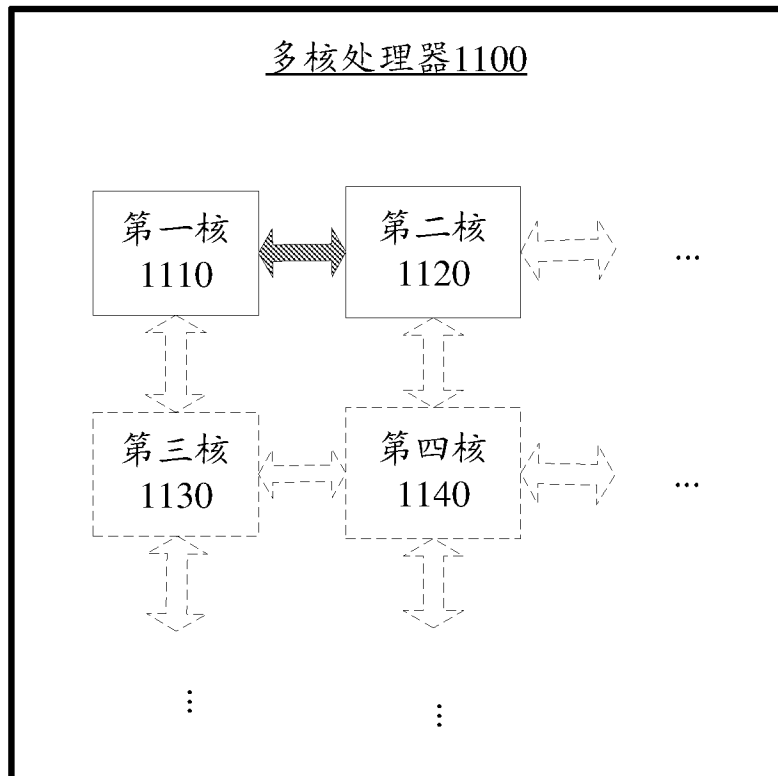


图 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/080867

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/00(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPODOC, CNPAT, CNKI, IIEEE: TLB, translation lookaside buffer, PA, physical address, physical frame number, PFA, CPU, core, missing, cache, buffer, full, mapping, replace

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 103119570 A (INTEL CORPORATION), 22 May 2013 (22.05.2013), description, paragraphs [0039]-[0050] and [0055]-[0059], and figure3	1-22
A	CN 105095094 A (HUAWEI TECHNOLOGIES CO., LTD. et al.), 25 November 2015 (25.11.2015), the whole document	1-22
A	CN 1848095 A (INTEL CORPORATION), 18 October 2006 (18.10.2006), the whole document	1-22
A	CN 104346294 A (HUAWEI TECHNOLOGIES CO., LTD. et al.), 11 February 2015 (11.02.2015), the whole document	1-22
A	US 6105113 A (SILICON GRAPHICS, INC.), 15 August 2000 (15.08.2000), the whole document	1-22

Further documents are listed in the continuation of Box C.

See patent family annex.

<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>	<p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&”document member of the same patent family</p>
---	--

Date of the actual completion of the international search
19January 2017 (19.01.2017)

Date of mailing of the international search report
08February 2017 (08.02.2017)

Name and mailing address of the ISA/CN:
State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao
Haidian District, Beijing 100088, China
Facsimile No.: (86-10) 62019451

Authorized officer
YANG, Dong
Telephone No.:(86-10) **010-62413693**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/080867

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103119570 A	22 May 2013	EP 2619675 A2 TW 201220046 A WO 2012040723 A2 US 2012079232 A1 CN 105760312 A KR 20130048261 A	31 July 2013 16 May 2012 29 March 2012 29 March 2012 13 July 2016 09 May 2013
CN 105095094 A	25 November 2015	WO 2015169145 A1	12 November 2015
CN 1848095 A	18 October 2006	US 2006143390 A1	29 June 2006
CN 104346294 A	11 February 2015	EP 3029575 A1 WO 2015014302 A1 US 2016147666 A1	08 June 2016 05 February 2015 26 May 2016
US 6105113 A	15 August 2000	None	

<p>A. 主题的分类</p> <p>G06F 12/00 (2006.01) i</p> <p>按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类</p>																				
<p>B. 检索领域</p> <p>检索的最低限度文献 (标明分类系统和分类号)</p> <p>G06F</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))</p> <p>WPI, EPODOC, CNPAT, CNKI, IEEE: TLB, translation lookaside buffer, 转址旁路缓存, 物理地址, PA, physical address, 物理框号, physical frame number, PFA, CPU, 内核, 核心, core, 未命中, missing, 缓存, buffer, 满, full, 映射, mapping, 替换, replace</p>																				
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>CN 103119570 A (英特尔公司) 2013年 5月 22日 (2013 - 05 - 22) 说明书第[0039]-[0050]、[0055]-[0059]段, 附图3</td> <td>1-22</td> </tr> <tr> <td>A</td> <td>CN 105095094 A (华为技术有限公司 等) 2015年 11月 25日 (2015 - 11 - 25) 全文</td> <td>1-22</td> </tr> <tr> <td>A</td> <td>CN 1848095 A (英特尔公司) 2006年 10月 18日 (2006 - 10 - 18) 全文</td> <td>1-22</td> </tr> <tr> <td>A</td> <td>CN 104346294 A (华为技术有限公司 等) 2015年 2月 11日 (2015 - 02 - 11) 全文</td> <td>1-22</td> </tr> <tr> <td>A</td> <td>US 6105113 A (SILICON GRAPHICS, INC.) 2000年 8月 15日 (2000 - 08 - 15) 全文</td> <td>1-22</td> </tr> </tbody> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	A	CN 103119570 A (英特尔公司) 2013年 5月 22日 (2013 - 05 - 22) 说明书第[0039]-[0050]、[0055]-[0059]段, 附图3	1-22	A	CN 105095094 A (华为技术有限公司 等) 2015年 11月 25日 (2015 - 11 - 25) 全文	1-22	A	CN 1848095 A (英特尔公司) 2006年 10月 18日 (2006 - 10 - 18) 全文	1-22	A	CN 104346294 A (华为技术有限公司 等) 2015年 2月 11日 (2015 - 02 - 11) 全文	1-22	A	US 6105113 A (SILICON GRAPHICS, INC.) 2000年 8月 15日 (2000 - 08 - 15) 全文	1-22
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求																		
A	CN 103119570 A (英特尔公司) 2013年 5月 22日 (2013 - 05 - 22) 说明书第[0039]-[0050]、[0055]-[0059]段, 附图3	1-22																		
A	CN 105095094 A (华为技术有限公司 等) 2015年 11月 25日 (2015 - 11 - 25) 全文	1-22																		
A	CN 1848095 A (英特尔公司) 2006年 10月 18日 (2006 - 10 - 18) 全文	1-22																		
A	CN 104346294 A (华为技术有限公司 等) 2015年 2月 11日 (2015 - 02 - 11) 全文	1-22																		
A	US 6105113 A (SILICON GRAPHICS, INC.) 2000年 8月 15日 (2000 - 08 - 15) 全文	1-22																		
<p><input type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p>																				
<p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p>																				
<p>国际检索实际完成的日期</p> <p>2017年 1月 19日</p>		<p>国际检索报告邮寄日期</p> <p>2017年 2月 8日</p>																		
<p>ISA/CN的名称和邮寄地址</p> <p>中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10) 62019451</p>		<p>授权官员</p> <p>杨栋</p> <p>电话号码 (86-10) 010-62413693</p>																		

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2016/080867

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	103119570	A	2013年 5月 22日	EP	2619675	A2	2013年 7月 31日
				TW	201220046	A	2012年 5月 16日
				WO	2012040723	A2	2012年 3月 29日
				US	2012079232	A1	2012年 3月 29日
				CN	105760312	A	2016年 7月 13日
				KR	20130048261	A	2013年 5月 9日
CN	105095094	A	2015年 11月 25日	WO	2015169145	A1	2015年 11月 12日
CN	1848095	A	2006年 10月 18日	US	2006143390	A1	2006年 6月 29日
CN	104346294	A	2015年 2月 11日	EP	3029575	A1	2016年 6月 8日
				WO	2015014302	A1	2015年 2月 5日
				US	2016147666	A1	2016年 5月 26日
US	6105113	A	2000年 8月 15日	无			