(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112119377 B
(45)授权公告日 2024.08.02

(21)申请号 201980032403.5
(22)申请日 2019.05.09
(65)同一申请的已公布的文献号
申请公布号 CN 112119377 A
(43)申请公布日 2020.12.22
(30)优先权数据
1808527.4 2018.05.24 GB
(85)PCT国际申请进入国家阶段日
2020.11.13
(86)PCT国际申请的申请数据
PCT/GB2019/051273 2019.05.09
(87)PCT国际申请的公布数据
WO2019/224518 EN 2019.11.28

(73)专利权人 ARM有限公司
地址 英国剑桥
(72)发明人 阿拉斯塔尔·大卫·瑞德
(74)专利代理机构 北京东方亿思知识产权代理有限责任公司 11258
专利代理师 田琳婧

(51)Int.Cl.
G06F 9/38(2006.01)
G06F 21/60(2006.01)
G06F 9/46(2006.01)
(56)对比文件
US 5978909 A,1999.11.02
审查员 王晓明

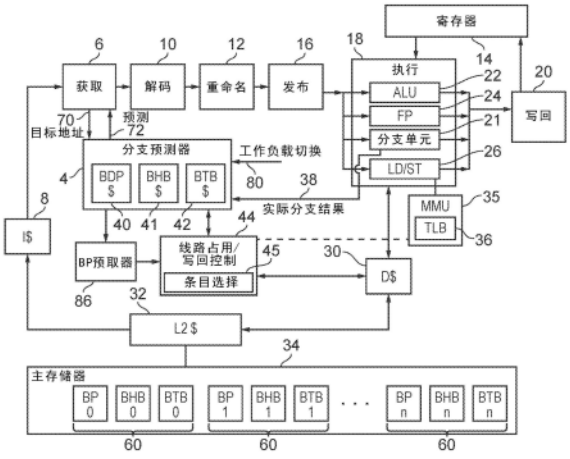
权利要求书3页 说明书17页 附图7页

(54)发明名称

用于多个软件工作负载的分支预测缓存

(57)摘要

一种装置,包括:处理电路(18),处理来自多个软件工作负载的指令;分支预测缓存(40-42),对选自存储器系统(30、32、34)中存储的多个分支预测状态数据集(60)的分支预测状态数据进行缓存,每个分支预测状态数据集对应于所述多个软件工作负载之一;以及分支预测电路(4),基于分支预测缓存中缓存的来自对应于给定软件工作负载的分支预测状态数据集的分支预测状态数据来预测给定软件工作负载的分支指令的结果。这对于减轻利用分支预测器的恶意训练导致的分支误预测的推测侧信道攻击是有用的。



CN 112119377 B

1.一种用于分支预测的装置,包括:

处理电路,处理来自多个软件工作负载的指令;

分支预测缓存,对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集与所述多个软件工作负载之一相对应;以及

分支预测电路,基于所述分支预测缓存中缓存的、来自与给定软件工作负载相对应的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果,

其中,所述分支预测缓存被配置为响应于所述处理电路从处理第一软件工作负载切换到处理第二软件工作负载,而阻止与所述第一软件工作负载相对应的分支预测状态数据集的缓存的分支预测状态数据被用于预测所述第二软件工作负载的分支指令的结果。

2.如权利要求1所述的装置,其中,所述分支预测缓存包括多个条目,每个条目用于缓存相应部分的分支预测状态数据,而与所述相应部分的分支预测状态数据选自哪个分支预测状态数据集的任意指示无关。

3.如权利要求1所述的装置,其中,所述分支预测缓存被配置为确保所述分支预测缓存的所有有效条目与来自与给定软件工作负载相对应的同一分支预测状态数据集的分支预测状态数据有关。

4.如权利要求1所述的装置,其中:

所述分支预测缓存被配置为响应于所述处理电路从处理第一软件工作负载切换到第二软件工作负载,来触发针对所述存储器系统的至少一个状态获取请求以请求将与所述第二软件工作负载相对应的分支预测状态数据集的部分获取到所述分支预测缓存中。

5.如权利要求4所述的装置,其中,所述分支预测状态数据集的部分包括与所述第二软件工作负载相对应的分支预测状态数据集的真子集。

6.如权利要求1所述的装置,其中,所述分支预测缓存响应于从所述分支预测缓存中逐出脏分支预测状态数据,来触发至少一个写回请求以请求将所述脏分支预测状态数据写回到所述存储器系统。

7.如权利要求1所述的装置,其中,所述分支预测缓存被配置为以与预定大小的地址范围相对应的粒度来缓存分支预测状态项信息,使得所述分支预测电路能够针对所述预定大小的相邻地址范围中的地址进行针对分支指令的结果的不同预测;并且

所述装置包括状态传输电路,该状态传输电路用于以给定大小为单位在所述分支预测缓存和所述存储器系统之间传输分支预测状态数据,每个单位包括用于针对至少一个指令地址块进行分支预测的分支预测状态数据,所述至少一个指令地址块中的每个指令地址覆盖的连续地址范围的大小大于所述预定大小。

8.如权利要求1所述的装置,包括分支预测状态预取器,该分支预测状态预取器将与预测将来需要分支预测的地址相关联的分支预测状态数据预取到所述分支预取缓存中。

9.如权利要求1所述的装置,其中,所述分支指令的结果包括所述分支指令的目标地址。

10.如权利要求1所述的装置,其中,所述分支指令的结果包括分支是否被采用。

11.如权利要求1所述的装置,其中,对于所述软件工作负载中的至少一个软件工作负载,针对给定软件工作负载的分支预测状态数据集被存储在所述给定软件工作负载不可访

问的存储器区域中。

12.如权利要求1所述的装置,其中,每个软件工作负载包括以下各项中的一项:

由所述处理电路执行的软件处理;

由所述处理电路执行的一组软件处理;

软件处理的特定部分;

具有特定地址范围中的指令地址的软件处理的指令;

由一个或多个工作负载划分指令界定的软件处理的一部分;以及

由所述处理电路执行的与同一处理相对应的多个线程之一,或所述多个线程中的线程子组。

13.一种用于分支预测的装置,包括:

处理单元,用于处理来自多个软件工作负载的指令;

缓存单元,用于对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集与所述多个软件工作负载之一相对应;以及

预测单元,用于基于所述缓存单元中缓存的来自与给定软件工作负载相对应的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果,

其中,所述缓存单元被配置为响应于所述处理单元从处理第一软件工作负载切换到处理第二软件工作负载,而阻止与所述第一软件工作负载相对应的分支预测状态数据集的缓存的分支预测状态数据被用于预测所述第二软件工作负载的分支指令的结果。

14.一种用于分支预测的方法,包括:

处理来自多个软件工作负载的指令;

在分支预测缓存中对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集与所述多个软件工作负载之一相对应;以及

基于所述分支预测缓存中缓存的来自与给定软件工作负载相对应的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果,

其中,响应于从处理第一软件工作负载切换到处理第二软件工作负载,而阻止与所述第一软件工作负载相对应的分支预测状态数据集的缓存的分支预测状态数据被用于预测所述第二软件工作负载的分支指令的结果。

15.一种用于分支预测的装置,包括:

处理电路,处理来自多个软件工作负载的指令;

分支预测缓存,对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集与所述多个软件工作负载之一相对应;以及

分支预测电路,基于所述分支预测缓存中缓存的、来自与给定软件工作负载相对应的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果,

其中,所述分支预测缓存响应于针对需要分支预测的给定软件工作负载的指令的目标指令地址在所述分支预测缓存中的未命中,来触发针对所述存储器系统的至少一个状态获取请求以请求将与所述给定软件工作负载相对应的分支预测状态数据集的相应部分获取到所述分支预测缓存中。

16.一种用于分支预测的装置,包括:

处理电路,处理来自多个软件工作负载的指令;

分支预测缓存,对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集与所述多个软件工作负载之一相对应;

分支预测电路,基于所述分支预测缓存中缓存的、来自与给定软件工作负载相对应的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果;以及

选择电路,响应于所述给定软件工作负载的给定指令的目标地址,选择来自与所述给定软件工作负载相对应的分支预测状态数据集的相应分支预测状态数据项,与所述给定软件工作负载相对应的分支预测状态数据集包括多个子表;

其中,所述选择电路被配置为基于所述目标地址的第一部分选择所述多个子表中的一个子表,并基于所述目标地址的第二部分的函数从所选择的子表中选择分支预测状态数据项。

用于多个软件工作负载的分支预测缓存

技术领域

[0001] 本技术涉及数据处理领域,更具体地涉及分支预测。

背景技术

[0002] 数据处理装置可以具有用于在分支指令被实际执行之前预测分支指令的结果的分支预测电路。通过在分支指令被实际执行之前预测分支结果,在分支指令的执行完成之前开始提取并推测性地执行跟随分支的后续指令,因此,在预测正确的情况下节省性能,因为一旦分支的结果实际已知,则后续指令可以比它们仅被提取时更快地被执行。

发明内容

[0003] 至少一些示例提供了一种装置,包括:处理电路,处理来自多个软件工作负载的指令;分支预测缓存,对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集对应于所述多个软件工作负载之一;以及分支预测电路,基于分支预测缓存中缓存的来自对应于给定软件工作负载的分支预测状态数据集的分支预测状态数据,来预测给定软件工作负载的分支指令的结果。

[0004] 至少一些示例提供了一种装置,包括:用于处理来自多个软件工作负载的指令的单元;用于对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存的单元,每个分支预测状态数据集对应于所述多个软件工作负载之一;以及用于基于分支预测缓存中缓存的来自对应于给定软件工作负载的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果的单元。

[0005] 至少一些示例提供了一种数据处理方法,包括:处理来自多个软件工作负载的指令;在分支预测缓存中对选自存储器系统中存储的多个分支预测状态数据集的分支预测状态数据进行缓存,每个分支预测状态数据集对应于所述多个软件工作负载之一;以及基于分支预测缓存中缓存的来自对应于给定软件工作负载的分支预测状态数据集的分支预测状态数据,来预测所述给定软件工作负载的分支指令的结果。

附图说明

[0006] 根据结合附图阅读的示例的以下描述,本技术的进一步方面、特征、和优点将显而易见,其中:

[0007] 图1示意性地示出了具有分支预测电路和分支预测缓存的数据处理装置的示例;

[0008] 图2示意性地示出了可以利用分支预测器诱使更多特权代码向攻击者泄露机密信息的潜在安全攻击;

[0009] 图3示出了用于缓存来自存储器系统中的备份存储装置的分支预测状态数据的部分的分支预测缓存的示例;

[0010] 图4示出了在工作负载之间切换并以相应方式在分支预测状态数据集之间切换的示例;

[0011] 图5A和5B示出了用于改善局部性的缓存组织的示例;

[0012] 图6是示出进行分支预测的方法的流程图;以及

[0013] 图7示出了响应于检测到的处理工作负载的切换来更新分支预测缓存的方法。

具体实施方式

[0014] 数据处理装置可以具有用于预测分支指令的结果的分支预测机构。例如,预测结果可以包括是否采用分支,或者可以包括分支指令的预测目标地址。还可以预测分支的其他特性,例如,特定分支是否与功能调用有关或者预期特定分支是否具有不可预测的分支地址模式,使得同一分支的不同执行实例可以产生不同的目标地址。传统上,分支预测机构将被认为是一种性能增强机构,该机构的误预测对于在数据处理装置中处理的数据的安全性不重要,而仅仅会影响所达到的性能等级。这是因为如果分支被错误预测,则在处理电路可以临时执行错误指令时,一旦确定了实际分支结果并将其与预测结果进行比较,即可检测到误预测,随后处理管线可以被基于误预测分支获取的后续指令刷新,并且处理器的架构状态可以被恢复到遇到分支的点,从而可以消除最后证明不正确的后续推测指令的架构效果。

[0015] 但是,最近已经认识到,分支预测电路可以提供攻击者可以用来规避安全保护的路线,其中,可以在数据处理装置上提供安全保护以禁止一些软件工作负载访问与其他软件工作负载相关联的数据。这是因为在一些分支预测器中,基于在一个软件工作负载中观察到的分支历史而分配给分支预测器的分支预测状态条目可以被从不同的软件工作负载访问并被用于预测不同软件工作负载的分支的结果。先前,使用来自一个工作负载的分支预测状态来预测另一工作负载中的分支的结果仅被认为是性能问题,如同第二工作负载命中由第一工作负载分配的分支预测器的错误条目,由此产生的任何误预测随后可以被识别出来并且可以在实际分支结果已知时被解析,预期这仅会导致处理正确分支结果的延迟而不会导致安全风险。

[0016] 但是,已经认识到,由于误预测的分支被不正确地推测性执行的指令仍然会影响缓存或数据处理装置使用的其他非架构存储结构中的数据。这可以被攻击者用来尝试获取有关潜在的敏感数据的一些信息,其中,该潜在的敏感数据对于攻击者是不可访问的但是对于另一软件工作负载是可访问的,该另一软件工作负载可以被攻击者诱使执行被设计为访问秘密并导致向攻击者暴露关于秘密的一些信息的缓存分配的改变。例如,攻击者可以利用分支访问的模式来训练分支预测器,使得在受害者软件工作负载随后访问同一条目时,其将不正确地执行来自错误目标地址的指令或遵循是否采用分支的错误预测,从而导致对秘密信息的不当访问。缓存定时侧信道随后可以被用来探测不正确的推测对泄露有关秘密信息的影响。

[0017] 解决这种攻击的一种方式可以是利用相应软件工作负载的标识符对分支预测器中的条目进行标记,使得给定软件工作负载仅可以使用标记有相应工作负载标识符的分支预测状态。但是,通常用在数据处理系统中来识别特定软件工作负载的标识符(例如,识别单个处理的处理标识符或地址空间标识符和/或识别给定处理在其中操作的虚拟机的虚拟机标识符)可以相对较长。例如,这种标识符可以包括多达32或64位。如果分支预测器的每个条目被标记有相应的软件工作负载标识符(或更糟地,诸如处理ID和虚拟机ID的标识符

集),则这会大大增加分支预测器的每个条目的大小。这对于预测是否采用分支的分支方向预测器尤其是个问题,因为对于这种分支方向预测器,每个分支预测状态项一般可以仅包括数个位(例如,用于跟踪预测是否应采用分支的置信度的2位置信度计数器)。因此,对每个条目进行标记将大大增加分支预测器的开销,这对于电路面积有限的微架构设计是不可接受的。标记的另一个问题在于,针对每个软件工作负载保持分支预测状态集,使得该软件工作负载的分支可以被充分预测,这将需要在分支预测电路中提供更大数目的条目来容纳每个工作负载的所有分支预测状态集(相比在所有工作负载之间共享相同条目的分支预测器)。因此,在实践中,利用每个工作负载的唯一软件工作负载标识符对条目进行标记并将分支预测器中的命中仅限制到对应于相同工作负载的条目在微架构设计方面可能是不实际的。

[0018] 在下面讨论的技术中,提供分支预测缓存来缓存从存储器系统中存储的两个以上分支预测状态数据集中选择的分支预测状态数据。每个分支预测状态数据集对应于由处理电路处理的多个软件工作负载之一。分支预测电路基于分支预测缓存中缓存的、来自对应于给定软件工作负载的分支预测状态数据集的分支预测状态数据,来预测给定软件工作负载的分支指令的结果。

[0019] 因此,通过在存储器系统中存储多个分支预测状态数据集,这允许支持各自具有自己的分支预测状态集的大量不同的软件工作负载,而在微架构中不需要过大的区域开销来支持分支预测电路局部的相应大小的分支预测缓存。即,存储器系统有效地充当用于存储所有软件工作负载的全套分支预测数据集的后备存储器,同时该分支预测状态数据的子集被缓存在分支预测缓存中供分支预测电路快速访问,使得缓存本身不需要具有比在工作负载之间共享分支预测状态的一般分支预测结构更大的电路面积预算。

[0020] 这种方法可以被看作是反直觉的,因为分支预测状态数据一般被看作通常将不会被暴露给存储器的给定处理管线的微架构方面。另外,预期与从存储器系统中的后备存储器获取分支预测状态数据相关联的延迟将比根本没有进行任何预测的情况下预期的延迟更长,而管线等待实际的分支指令结果在获取后续指令之前可用于执行。因此,存储器系统将不可能被认为是用于存储分支预测状态数据的适当位置。然而,在实践中,软件工作负载中的变化会相对少见,在开启新软件工作负载时获取分支预测状态数据的预热延迟在任何情况下都可以与在工作负载之间共享分支预测器条目的分支预测器中的预热延迟相比较,因为先前针对一个工作负载所做的预测逐渐响应于针对第二工作负载的错误预测的分支进行适应。相反,通过使用存储器系统中存储的不同分支预测状态数据集,基于一个工作负载中的实际分支结果的校正不会污染另一工作负载使用的分支预测状态,因此可以使在它们被携带到缓存中时对于这两个工作负载的预测更加精确,从而有助于减轻与从存储器系统获取分支预测状态数据相关联的一些延迟。在存储器系统中存储不同的分支预测状态数据集的另一优点在于,在多处理器系统中,当给定软件工作负载被从一个处理器移到另一处理器时,如果分支预测状态数据被存储在存储器中,则意味着分支预测状态也可以相对容易地被从一个处理器迁移到另一个处理器,而不需要建立专门的分支预测状态传输路径,这可以简化处理器系统设计并提高性能(在调度器在处理器之间移动软件工作负载时,通过避免推测性能的下降)。因此,在实践中,可以限制性能影响,并且通过在存储器系统中保持用于不同软件工作负载的不同分支预测状态数据集,可以阻止操作一个软件工作负载

的攻击者能够使用其使用自己的工作负载进行训练的分支预测状态数据,从而导致被设计为向该攻击者暴露有关秘密数据的信息的另一软件工作负载中的不正确的推测。

[0021] 响应于处理电路从处理第一软件工作负载切换到处理第二软件工作负载,分支预测缓存可以阻止对应于第一软件工作负载的分支预测状态数据集中缓存的分支预测状态数据被用来预测第二软件工作负载的分支指令的结果。例如,在不同软件工作负载之间的上下文切换上,分支预测缓存可以使与第一软件工作负载相关联的条目无效。如果任意无效条目包含相对于存储器系统中的相应预测状态改变的更新后的预测状态数据,则这种脏分支预测状态数据可以被写回到存储器系统。可以在第一软件工作负载和第二软件工作负载之间的上下文切换时执行写回,或者写回可以被推迟到期望利用与第二软件工作负载相关联的新分支预测状态数据覆写给定缓存条目为止。因此,一般通过更新缓存状态使得与第一软件工作负载相关联的分支预测状态数据无法被用来预测第二软件工作负载中的分支指令的结果,避免了上面讨论的类型的安全攻击。

[0022] 分支预测缓存可以具有多个条目,每个条目用于缓存分支预测状态数据的相应部分,而与该分支预测状态数据的相应部分所选自哪个分支预测状态数据集的任意指示无关。即,分支预测缓存的每个条目不需要包括对应于该条目中的分支预测状态数据的特定软件工作负载的任意标识符。通过避免利用与软件工作负载相关联的处理标识符或其他标识符对条目进行标记,避免了预期由这种标记产生的面积开销的大量增加。这对于分支目标地址预测器和分支方向预测器二者可以是有利的,但是可以为分支方向预测器提供特定好处(因为每个分支预测状态数据项的大小通常较小,所以利用处理标识符进行标记将使得每个条目的大小不成比例地增加)。

[0023] 分支预测缓存可以确保分支预测缓存的所有有效条目都与来自对应于给定软件工作负载的同一分支预测状态数据集的分支预测状态数据有关。这避免了需要在每个条目中包括区分给定条目所涉及的软件工作负载的标记。当软件工作负载发生变化时,可以替代地通过使分支预测缓存的内容无效来确保阻止与一个软件工作负载相关联的条目被用来进行针对另一软件工作负载的预测。可以根据指令地址进行针对分支预测缓存的查找,但与当前工作负载的工作负载标识符无关。当前工作负载的身份可以被认为在分支预测缓存线占用上而不在每个预测查找上。

[0024] 注意,当分支预测缓存的条目被无效时,这并不意味着存储器中的相应分支预测状态数据需要被无效。相反,存储器中的分支预测状态数据可以被保留并且在必要时可以基于从分支预测缓存逐出的数据被更新,使得分支预测状态数据可以持续超过执行给定软件工作负载的一个实例并且在下次执行同一软件工作负载时被恢复。

[0025] 响应于处理电路从处理第一软件工作负载切换到第二软件工作负载,分支预测缓存可以触发针对存储器系统的至少一个状态获取请求,以请求将对应于第二软件工作负载的分支预测状态数据集的一部分获取到分支预测缓存中。因此,与输入软件工作负载相关联的新分支预测状态数据集的加载可以由从第一软件工作负载到第二软件工作负载的上下文切换自动触发。

[0026] 在一些情况下,在上下文切换时获取到分支预测缓存中的分支预测状态数据集的部分可以包括与第二软件工作负载相关联的整个分支预测状态数据集。当遇到后续分支时这种方法可以减少对于处理器管线的性能影响,因为这意味着在上下文切换后尽可能早地

获取了相关分支预测状态数据。

[0027] 但是,在其他情况下,响应于从第一软件工作负载到第二软件工作负载的切换而获取的分组预测状态数据集的部分可以包括对应于第二软件工作负载的分支预测状态数据的真子集。即,并不是与第二软件工作负载相关联的所有分支预测状态数据集可以响应于上下文切换被获取到分支预测缓存中。相反,最初可以获取较小部分,然后分支预测状态数据的剩余部分可以在需要给定分支指令时按需要以较小的块被获取到分支预测缓存中。这种方法可以避免在上下文切换时占用太多存储器访问带宽(否则会不利地影响输入的第二软件工作负载进行的实际数据存储器访问的性能)。

[0028] 分支预测缓存可以响应于需要分支预测的给定软件工作负载的指令的目标指令地址在分支预测缓存中的未命中,触发针对存储器系统的至少一个状态获取请求,以请求将对应于给定软件工作负载的分支预测状态数据集的相应部分获取到分支预测缓存中。因此,如果遇到需要分支预测的目标指令地址,则这随后可以触发获取分支预测状态数据相应部分(如果其尚不存在于分支预测缓存中)。通过将给定软件工作负载的分支预测状态集分割成由分支预测缓存临时缓存但是可以被写回到存储器系统的较低等级的多个块,这减小了上下文切换的延时并避免了对于总状态大小的限制。

[0029] 在其他实施方式中,可以不响应于上下文切换本身触发任何状态获取请求,而可以根据需要来获取所有分支预测数据,使得只有在遇到需要分支预测状态数据的地址时才将分支预测状态数据的给定部分获取到分支预测缓存中。

[0030] 在实践中,在响应于上下文切换获取分支预测状态数据的初始部分并且根据需要稍后获取后续部分的情况下,这两种方法之间的平衡可能是最高效的。

[0031] 当分支预测状态数据被从分支预测缓存中逐出时(在分支预测状态数据被无效时的上下文切换,或者在给定分支预测状态数据随后基于分支预测状态数据的按需获取被覆写时的上下文切换),如果逐出的数据是脏数据(相对于存储器中的相应分支预测状态数据已经改变),则分支预测缓存可以触发至少一个写回请求,以请求将脏分支预测状态数据写回到存储器系统。这种写回对于分支预测器来说通常是不需要的,因为通常期望分支预测器将分支预测状态的完整记录保留在处理管线设计的微架构中,而不是充当存储器系统中的后备存储器的缓存。

[0032] 典型的分支预测器通常被设计为具有存储布置,由此用于预测给定目标指令地址处的分支的结果的分支预测器的条目基于应用于目标指令地址的哈希算法而被识别,其中,该哈希算法被设计为确保相邻地址映射到分支预测器的不同条目。即,分支预测器(包括分支方向预测器和分支目标地址预测器二者)通常被设计为具有非常小的局部性,以避免在程序代码的同一区域中遇到的多个分支争用对同一分支预测条目的访问的热点。

[0033] 但是,发明人认识到,如果这种典型的分支预测器哈希方案被应用于上面讨论的分支预测缓存,则这可能会在分支预测缓存和存储系统之间传输分支预测状态数据项时提供非常差的性能。这是因为,当给定软件工作负载处理程序代码的给定部分时,利用传统的分支预测映射方案,相邻分支的地址将被映射到随后可能需要多个单独获取操作的分支预测状态数据集的非常不同的条目上,以将它们从存储器系统加载到分支预测缓存。因此,利用缓存来自存储器系统的分支预测状态数据子集的分支预测缓存,对于提供这样的地址映射方案可能是有用的,其中,该地址映射方案相比于传统的分支预测哈希方案提供了更大

的局部性。

[0034] 分支预测缓存可以按照对应于预定大小的地址范围的粒度缓存分支预测状态信息项。这意味着,分支预测电路能够针对预定大小的相邻地址范围中的地址进行分支指令结果的不同预测。即,预定大小可以是可以从相同大小的下个地址范围进行单独预测的最小地址范围。

[0035] 该装置可以具有状态传输电路,该状态传输电路以给定大小为单位在分支预测缓存和存储器系统之间传输分支预测状态数据,其中,每个单位包括针对至少一个指令地址块进行分支预测的分支预测状态数据,并且每个这样的指令地址块覆盖的连续地址范围的大小大于用作分支预测缓存的粒度的预定大小。因此,分支预测状态可以在一次数据传输中被传输到分支预测缓存的地址范围可以比分支预测缓存中用来表示单独预测的粒度更大。这可以提高存储器访问性能,因为其可以减少加载用于预测程序代码的给定部分的分支预测结果的分支预测状态所需要的存储器访问数目。

[0036] 在一个示例中,该装置可以包括选择电路,该选择电路响应于给定软件工作负载的给定指令的目标地址,从对应于给定软件工作负载的分支预测状态数据集中选择相应的分支预测状态数据项。例如,选择电路可以是用于选择响应于目标地址查找的分支预测缓存的条目的查找电路,或者可以被包括在用于在分支预测缓存中存在未命中或者软件工作负载之间存在上下文切换时将从其获取所需要的分支预测状态的存储器地址的电路(在实践中,查找电路和存储器地址生成电路可以共享相同的部分)。在一些示例中,除了目标地址外,选择电路还可以基于指示先前的分支结果的模式的分支历史信息来选择分支预测状态数据项(例如,多个最近执行的分支的采用结果或不采用结果),这可以帮助提供更精确的预测(因为针对通过程序去往同一分支采用的不同线路可以生成不同的预测)。

[0037] 对应于给定软件工作负载的分支预测状态数据集可以包括多个子表,选择电路可以基于目标地址的第一部分选择一个子表。可以基于目标地址的第二部分的函数从所选择的子表中选择用于提供单独预测的分支预测状态数据项。每个子表可以提供对应于覆盖比上面讨论的预定大小更大的连续地址范围的至少一个指令地址块的分支预测状态数据。因此,子表的选择使能可以在一次数据传输中传输到分支预测缓存中的分支预测状态数据单元的选择,随后可以基于目标地址的剩余部分的函数(例如,类似于传统分支预测器中使用的哈希函数的哈希函数,但是基于目标地址的较小部分,以减少子表中的局部性并减少热点)从所选择的子表中选择单独的分支预测状态数据项。

[0038] 换言之,利用使用子表的这种方法,每个子表可以充当分支预测缓存的单独缓存条目,但是可以在该缓存条目中包括针对预定大小的不同地址范围的多个单独预测状态项。通过将每个缓存条目视为地址空间的某个子集的独立子表,增加了局部性,从而提高了对于分支预测缓存的存储器访问性能。

[0039] 注意,给定子表可以被映射到非连续位置处的多个不同地址块,但是每个块覆盖的连续地址范围的大小大于预定大小。用来选择子表的选择函数可以在多个不连续地址块之间共享单个子表,以期望在程序代码执行来自这些不连续块中的一个块的指令时由分支预测器进行的预测在同一工作负载随后到达这些不连续块中的另一块时不再相关,使得先前针对第一代码区域进行的预测可以被针对下一区域进行的预测覆写。这可以避免需要针对给定软件工作负载提供很多子表,从而限制了用于存储分支预测状态数据集的存储器容

量。

[0040] 用来选择子表的目标地址的第一部分可以是比用来选择子表中的单独分支预测状态数据项的第二部分的至少一部分更高有效的部分。在一些情况下,第一部分可以是目标地址的最高有效部分。但是,在其他示例中,第一部分可以是目标地址的中间部分,因此目标地址的第二部分不仅可以包括没有第一部分高效的部分,而且还可以可选地包括比第一部分更高有效的一个或多个位。

[0041] 可以提供分支预测状态预取器,以将与预测未来需要分支预测的地址相关联的分支预测状态数据预取到分支预测缓存中。分支预测状态预取器可以类似于基于数据访问地址进行训练的硬件数据预取器操作,以预测未来的数据地址访问模式,例如,所访问的地址以规则偏移模式递增的步长模式。可以基于供给到分支预测器的指令地址以类似方式训练分支预测状态预取器,使得在程序代码逐步执行指令序列的情况下,分支预测状态可以在获取阶段达到的当前程序执行点之前,预取即将预测达到的指令的分支预测状态数据(在获取阶段实际达到这些指令之前)。这可以有助于通过减少与获取当前指令的相关分支预测状态数据相关联的延迟来提高性能。

[0042] 本申请中讨论的技术可以应用于分支目标缓冲器(也称为分支目标地址预测器),该分支目标缓冲器预测分支指令的目标地址,作为分支指令的结果。分支的目标地址是分支将程序流重定向到的地址。替代地,本技术可以应用于分支方向预测器,针对该分支方向预测器的分支指令的预测结果是是否应该采用分支的指示。将理解的是,上面讨论的技术可以仅应用于这些类型的分支预测结构中的一种类型或者可以应用于同一系统中的两种类型的分支预测结构。

[0043] 分支预测状态数据可以被存储在存储器系统的可寻址区域中,至少一些软件工作负载可以通过发布由指定映射到存储分支预测状态数据的区域的地址的处理电路执行的加载/存储指令来访问存储器系统的该可寻址区域。因此,不是将数据存储在对于由处理电路执行的指令可直接访问的微架构数据存储器中,分支预测状态数据可以被存储在处理电路访问的常规存储器的可寻址区域中。例如,操作系统可以在分配给给定处理的存储器区域中分配一些额外空间来存储该处理的分支预测状态。

[0044] 但是,对于在处理电路上执行的至少一些软件工作负载,用于给定软件工作负载的分支预测状态数据集可以被存储在对于该给定软件工作负载不可访问的存储器区域中。例如,可以针对存储相应分支预测状态数据集的区域指定存储器保护属性,以将对分支预测状态数据的访问限制到比其分支预测状态数据正在被存储的给定软件工作负载特权更高的处理。例如,给定处理的分支预测状态数据对于该处理是不可访问的,但是对于监管该处理的操作系统是可访问的。替代地,在一些情况下,该系统可以被实现为使得分支预测状态数据集对于在处理器上执行的任意软件工作负载都是不可访问的(即使它们被分配给通常可访问的存储器的区域),因此只有被允许访问可寻址存储器区域的系统元件可以是与控制分支预测状态的获取和写回的分支预测器相关联的元件。

[0045] 在分支预测状态数据通过处理电路执行的通用负载存储指令对于至少一些软件工作负载可见的情况下,可以使相应的存储器区域对于这些指令只读,以阻止攻击者试图产生上面讨论的攻击形式而将不正确的分支预测状态数据写到存储器系统中的能力。例如,被允许写入存储用于给定软件工作负载的分支预测状态数据集的存储器区域的唯一系

统元件可以是分支预测缓存或分支预测电路本身(或用于控制分支预测状态数据从分支预测缓存的写回的任何相应逻辑)。

[0046] 每个软件工作负载可以包括以下各项中的一者:由处理电路执行的软件处理;由处理电路执行的一组软件处理;软件处理的特定部分;具有特定地址范围中的指令地址的软件处理的指令;由一个或多个工作负载划分指令界定的软件处理的一部分;由处理电路执行的对应于同一处理的多个线程之一;或所述多个线程中的线程子组。

[0047] 因此,准确地说,在如何将由处理电路处理的线程或处理指派给分别具有相应的分支预测状态数据集的不同软件工作负载方面存在灵活性。在一些情况下,如果一组软件处理被认为没有为彼此带来上面讨论的攻击形式方面的风险,则它们可以被映射到同一分支预测状态数据集,而被认为导致风险的其他软件处理或线程可以被指派给具有不同的分支预测状态数据集的不同工作负载。在一些情况下,软件处理的特定部分可以被映射到不同的分支预测状态集(即使在同一软件处理中)。例如,互联网浏览器可以将不同的浏览器选项卡分割为具有它们自己的分支预测状态数据集,以避免选项卡相互影响。类似地,单个处理使用的不同软件库可以被映射到不同的分支预测状态集。在软件处理的不同部分被认为是不同的软件工作负载的情况下,这些部分之间的划分可以在特定地址范围方面进行,使得具有特定地址范围中的指令地址的软件处理的指令可以被认为是一个软件工作负载,同时在不同地址范围中的指令可以被映射到具有不同的分支预测状态数据集的不同软件工作负载。另一选项是,工作负载划分指令可以被提供以标记给定软件处理在工作负载之间的分割点。

[0048] 图1示意性地示出了具有包括多个管线阶段的处理管线的数据处理装置2的示例。该管线包括用于预测分支指令的结果并生成要被获取的指令的一系列获取地址的分支预测器4。获取阶段6从指令缓存8获取由获取地址识别的指令。解码阶段10对所获取的指令进行解码,以生成用于控制管线的后续阶段的控制信息。重命名阶段12执行寄存器重命名,以将由指令识别的架构寄存器指示符映射到识别硬件中提供的寄存器14的物理寄存器指示符。寄存器重命名对于支持无序执行可以是有用的,因为这可以通过将它们映射到硬件寄存器文件中的不同物理寄存器来消除指定同一架构寄存器的指令之间的冲突,从而增加指令可以按照不同于它们的程序顺序(其中,从缓存8获取这些指令的顺序)的顺序被执行的可能性,这可以通过允许较晚的指令在较早的指令等待操作数变得可用的同时执行来提高性能。将架构寄存器映射到不同物理寄存器的能力还可以帮助架构状态在分支误预测情况下的回滚。发布阶段16对等待执行的指令进行排队,直到处理这些指令所需要的操作数在寄存器14中可用为止。执行阶段18执行指令以执行相应的处理操作。写回阶段20将所执行指令的结果写回到寄存器14。

[0049] 执行阶段18可以包括多个执行单元,诸如用于评估分支指令是否被正确预测的分支单元21、用于执行算术或逻辑运算的ALU(算术逻辑单元)22、用于使用浮点操作数执行运算的浮点单元24、和/或用于执行将数据从存储器系统加载到寄存器14的加载操作或将来自寄存器14的数据存储到存储器系统的存储操作的加载/存储单元26。在本示例中,存储器系统包括级1指令缓存8、级1数据缓存30、在数据和指令之间共享的级2缓存32、以及主存储器34,但是将理解的是,这仅仅是可能的存储器层次结构的一个示例,并且其他实施方式可以具有更多级的缓存或不同布置。可以使用用于控制地址转换和/或存储器保护的存储器

管理单元(MMU)35来控制对存储器的访问。加载/存储单元26可以使用MMU35的转换后备缓冲器36来将由管线生成的虚拟地址映射到识别存储器系统中的位置的物理地址。将理解的是,图1所示的管线仅是一个示例,其他示例可以具有管线阶段或执行单元的不同集合。例如,有序处理器可以不具有重命名阶段12。

[0050] 分支预测器4可以包括用于预测分支指令的各种结果的结构。例如,分支预测器4可以包括预测是否应该采用条件分支的分支方向预测器。分支方向预测器可以为相对较小的结构,因为对于给定指令地址或指令地址块,每个条目可以为置信度计数器提供相对较少的位(例如,2或3位),该置信度计数器在这些预测中的置信度增大时朝向“采用”指示或“不采用”指示递增或递减,并且在发生误预测时被调整以反转先前的置信度增加。可以预测的分支结果的另一方面可以是分支的目标地址。例如,一些分支指令基于寄存器14中存储的值来间接地计算目标地址,因此可以分支到根据程序代码本身不能确定性地获知的地址。分支目标缓冲器(BTB)(也称为分支目标地址缓存(BTAC))可以是分支预测器4的一部分,该分支预测器4具有分别提供在给定指令块中出现的任意分支的目标地址的预测的多个条目。可选地,BTB或BTAC还可以提供关于分支的其他信息,例如,特定类型的分支的预测(例如,功能调用、功能返回等)。另外,可以基于针对执行阶段的分支单元21所执行的分支指令确定的实际分支结果38来改进BTB/BTAC所进行的预测。可以包括在分支预测器4中的另一结构可以是分支历史缓冲器(BHB),该分支历史缓冲器可以记录关于先前执行的分支的历史的信息(例如,先前采用/不采用的结果的序列),该信息可以被用作用于生成例如对于BTB的索引的附加输入。

[0051] 图1所示的处理管线可以支持多个不同软件工作负载的执行。软件工作负载可以包括根据不同程序代码执行的不同处理,或者可以包括对应于同一处理的多个线程。另外,在一些情况下,处理中的不同部分可以被视为不同的工作负载,例如,处理中的某个地址范围可以被标记为单独的工作负载。

[0052] 当不同处理在同一管线上执行时,分支预测器4通常在这些处理之间共享。由于不同处理可以在同一指令地址处具有不同的分支行为,这可意味着查找给定指令地址的分支预测器结构可以提供与一个处理不相关的预测行为,因为它已经基于另一处理被训练。一般地,由一个处理访问由另一处理训练的分支预测条目而产生的分支误预测将仅被视为影响性能而不影响安全的问题,因为如果预测不正确,则其将会在分支在分支单元21中被实际执行时被检测出来并且随后分支单元可以触发管线以被基于误预测不正确地获取的后续指令而刷新,并且处理器状态可以重新回到由最后一个正确预测的指令产生的最后一个正确状态。

[0053] 然而,尽管误预测的架构效果可以被反转,但是误预测也可导致对于诸如数据缓存30或TLB 36的微架构状态的更持久的影响。最近已经认识到,攻击者有可能利用分支预测器4来访问攻击者不应该访问的秘密信息。存储器管理单元35可以应用特权方案,使得仅在某些特权等级执行的处理被允许访问某些存储器区域。例如,如图2所示,一些秘密数据50对于攻击者的处理可能是不可访问的(例如,因为攻击者的处理在最低特权等级运行),但是对于诸如操作系统或管理程序的在较高特权等级操作的处理可以是可访问的。秘密数据可以是诸如密码、个人财务资料等的被认为敏感的任何数据。攻击可以基于训练分支预测器4,使得在处理器的更高特权状态执行的受害者代码中的分支可以分支到更高特权的

受害者代码不希望执行但是由于受害者代码的分支中与秘密50不相关的分支误预测而不正确地执行的一些小工具代码52。小工具代码可以被攻击者设计为访问基于秘密数据50计算出的存储器地址,使得与取决于秘密数据的存储器地址相关联的数据(或诸如TLB条目的其他信息)被加载到数据处理系统的缓存30、32、36中的一者。

[0054] 更具体地,攻击者可能先前执行了确保条目54被分配给分支预测器4的分支目标缓冲器的指令模式,因此当分支指令地址#b处的指令被执行时,BTB返回对应于小工具代码52的位置的目标地址#g的预测。分支指令地址#b可以由攻击者选择,使得其对应于受害者代码中的特定分支指令56的地址(替代地,#b可以是受害者代码的非分支指令的地址,因为某些形式的分支预测器4除了预测分支的实际目标地址以外,还可以有效地预测指令是否是分支指令,所以错误地训练分支预测器以预测非分支指令为分支到#的分支指令也可以达到相同的效果)。

[0055] 因此,在受害者代码执行时其到达地址#b处的指令56,并且在分支预测器4被查找时其可以返回目标地址#g,使得更高特权受害者代码切换到地址#g处的小工具代码52。由于处理器仍然处于与受害者代码相关联的更高特权状态,因此攻击者所提供的小工具代码52现在执行具有与提供给受害者代码相同的访问权限的指令,这可以允许访问对于攻击者的较低特权代码不可访问的存储器区域。小工具代码可以包括加载秘密数据50的第一加载指令58和从作为第一加载指令58加载的秘密的函数计算得出的目标地址加载数据的第二加载指令60。攻击者可能选择了这种地址计算函数,所以根据秘密的值(或秘密的选定位的值),计算得出的地址可以采用图2的左手侧所示的若干值62、64中的一者。在本示例中,为了简洁,示出了计算得出的地址的两个可能值62和64,例如,这可以基于计算得出的地址取决于从秘密提取的单个位的函数。将理解的是,当基于秘密的更多位计算该函数时,计算得出的地址可以采用两个以上不同值。因此,取决于秘密的值,与不同地址相关联的信息可以被从存储器加载,并且这会使得条目针对地址62、64中的一者被分配在缓存结构30、32、36中。

[0056] 针对地址62、64之一的新条目的分配还会使其他条目被从缓存30、32、36之一中逐出。缓存结构通常可以被按照组关联方式实现,这意味着给定地址的条目仅可以被分配给有限位置集中的一个位置而不能被分配给缓存结构中的任意位置。索引到同一集合中的一组地址可以是预先已知的,或者可以通过分析访问不同地址模式所花费的时间来推断,以推断哪些地址倾向于与缓存资源冲突,从而导致访问延时增加。因此,攻击者可以能够从缓存访问定时推断出从缓存中逐出了哪个地址,以为第二加载60所加载的信息腾出空间,从而确定潜在的候选地址62和64中的哪个地址是基于秘密计算得出的。即,通过测量对于多个地址的缓存访问定时,攻击者可以推断出有关秘密的一些信息。尽管该信息可能不足以完全识别出秘密,但是通过多次重复攻击(例如,利用使用秘密的不同部分的不同地址计算函数),攻击者可以逐渐地将有关秘密的更多信息拼凑在一起从而推导出攻击者不应该被允许访问的一些敏感数据。

[0057] 图2仅示出了使用分支预测器来攻击秘密信息并将其暴露给特权较低代码的可能方式的一个示例。这种形式的攻击利用以下事实,即具有不同安全需求和信任等级的很多不同执行上下文通常可以共享单个分支预测器状态集,这允许攻击者影响另一处理的执行并监控另一处理的分支模式。还为攻击者提供了通过改变分支预测状态产生大量误预测分

支来降低其他处理的性能的方式。

[0058] 对于这些类型的攻击的一种可能的缓解方法可以是每当发生上下文切换时从分支预测器刷新分支预测状态,但是这种方法会比较昂贵,因为每当给定处理返回另一执行时隙时,它可能必须再次从头开始获取其分支预测状态,这可能会导致许多其他错误预测影响性能。另一种方式可以是利用处理标识符对每个分支预测状态项进行标记(在分支方向预测器或BTB中),使得它们仅在由当前执行的相同处理分配的情况下被使用。然而,这在电路面积和功率方面也会比较昂贵(由于唯一地识别需要不同的分支预测状态集的每个单独软件工作负载),这可能需要具有相对较大数目的位的标识符。在实践中,分支预测状态条目可以相对较小,尤其是对于单独分支预测条目可以为2位小的分支方向预测器来说。因此,向分支预测器的每个条目添加附加的工作负载标识符会使得分支预测结构所需要的存储容量和用于比较当前的工作负载标识符与分支预测条目中的标识符的相应比较逻辑大大地增加。另外,这种方法还需要在分支预测器中提供更大总数目的条目,使得其可以在同一微架构结构中容纳针对不同软件工作负载的多个独立分支预测状态集。

[0059] 在下面讨论的方法中,不在分支预测器4本身中存储所有的分支预测状态,而可以在存储器系统30、32、34中存储对应于各个软件工作负载的分支预测器状态集60,其中,存储器系统30、32、34可以为在分支预测器4的微架构中提供的较小的分支预测缓存结构40、41、42提供备份存储器。每个分支预测状态集60可以对应于特定的软件工作负载或者可以是对应于同一处理的多个线程中的单独线程,其中,该软件工作负载可以是单独处理、一组处理、处理中处于特定地址范围处的部分代码、给定处理中存在于连续工作负载划分指令之间的代码的部分。线路占用和写回控制电路44可以控制分支预测状态在分支预测缓存40-42与存储器系统30、32、34之间的传输。

[0060] 尽管在图1中为了简洁仅在主存储器34中示出了主存储器中的分支预测状态集60,但是将理解的是,该数据的多个部分也可以被分配到缓存30、32。分支预测状态集60可以被存储在由例如,诸如操作系统或管理程序的监管处理分配给相应的软件工作负载的存储器区域中。在一些情况下,分支预测状态集60可以被分配给对于相应的软件工作负载本身来说不可访问的存储器区域,使得给定的工作流不能看到其自身的分支预测数据。可以使存储分支预测数据集60的存储器区域仅对于在负载存储单元26中执行的指令可读,因此只有被准许写入分支预测状态集的系统的元件可以是线路占用/写回控制电路44,以避免攻击者能够损坏给定的软件工作负载的分支预测状态。因此,通过在存储器系统30、32、34(在RAM中)中存储分支预测状态(其可以包括分支方向预测状态BP、分支历史缓冲器状态BHB、以及分支目标缓冲器状态BTB中的一者、两者、或全部),可以支持大量处理或软件工作负载分别具有它们自己的分支预测状态集。在一些情况下,不同的软件库或不同的浏览器选项卡可以具有它们自己的分支预测状态集。分支预测缓存结构40-42可以被布置为确保在管线正在执行来自给定软件工作负载的指令时,只有来自分支预测状态集60的对应于该工作负载的分支预测状态可以被用来进行预测,这减轻了图2所示的攻击形式(因为攻击者的与一个分支预测状态集60相关联的处理不再可能以影响利用不同的分支预测状态集执行的另一处理的方式训练分支预测状态)。尽管图1示出了所有BDP、BHB、和BTB充当主存储器中的备份存储器中存储的分支预测状态的缓存的示例,但是这不是关键,并且在其他示例中,来自这些结构中的一个或多个结构的状态(例如,BHB)可以简单地在工作负载切换时

被丢弃而不会被保存在存储器中,同时其他形式的分支预测状态仍然可以被存储到存储器34。

[0061] 图3更详细地示出了分支预测缓存结构40-42的示例。为了简洁,这些结构在下面将都被称为分支预测缓存。将理解的是,分支预测缓存可以是缓存预测分支采用/不采用结果的信息的分支方向预测缓存40,或缓存对应于分支历史缓冲器的分支历史信息的部分的BHB缓存41、或缓存分支的目标地址的预测的BTB缓存42。

[0062] 分支预测缓存40-42具有分别提供针对相应指令地址块的预测的多个条目64。每个条目可以具有指示该条目是否提供有效数据的有效标志66、指示条目64提供的预测自从该条目被分配但是尚未被写回存储器34中存储的相应分支预测状态集60是否改变的脏标志68、用于识别与特定地址有关的条目的地址标志70、以及提供相应地址块的预测分支结果的预测状态72。将理解的是,缓存布置可以不同于图3所示的布置,并且每个条目可以包括图3未示出的其他形式的信息。

[0063] 如图3所示,分支预测缓存40-42(或统称为分支预测器4)可以具有到获取阶段6和执行阶段18的接口、以及经由线路占用/写回控制电路44到存储器系统30、32、34的接口。可以基于从获取阶段6提供的目标指令地址70查找分支预测缓存,并且在分支预测缓存中存在对应于该目标指令地址的条目的情况下,该缓存向获取阶段返回相应分支结果的预测72,该预测被用来控制从指令缓存8中获取后续指令。

[0064] 如果给定目标指令地址70在缓存中未命中(当前不存在对应于该指令地址的有效条目),则可以向线路占用/写回控制电路44发布线路占用请求74,该线路占用/写回控制电路可以具有从对应于当前软件工作负载的分支预测状态集60对存储相关分支预测状态项的相关存储器地址进行计算,然后向存储器系统发布将相关条目获取到缓存中的请求的条目选择电路45。在一些情况下,请求将相关分支预测状态加载到分支预测缓存40-42中的存储器访问请求可以经由加载存储单元26和/或MMU 35行进。替代地,线路占用/写回控制电路44可以具有到存储器的单独接口。分支预测状态可以被指派给物理地址或虚拟地址,并且在它们被指派给虚拟地址的情况下,可能需要使用TLB 36的地址转换来生成存储相关分支预测状态的实际物理地址。

[0065] 最终,响应于线路占用请求,线路占用获取响应76被从提供所请求的分支预测状态数据的存储器系统返回,随后其可以被分配到分支预测缓存40-42的给定条目中。如果必须逐出其他分支预测状态数据为新分配的条目腾出空间,则写回请求78可以被发送给存储器系统(如果如脏标志68所示,逐出条目中的数据为脏的)。在多处理器系统中,如果多个处理器共享同一分支预测数据集,则在一些示例中,可以应用一致性方案来确保由一个处理器触发的分支预测状态的写回对于在其他处理器的分支预测缓存40-42中缓存了同一分支预测数据的其他处理器可见。但是,确保一个处理器中的分支预测状态与相应分支预测状态之间的一致性并不是关键,因为在实践中,如果分支预测状态变得不一致,害处仅是误预测而不是错误的处理结果,所以其他实施方式将偏向保证多处理器系统的不同处理器中缓存的分支预测状态的一致性,以节省电路面积和复杂性。

[0066] 当指令基于分支预测器4上所做的分支预测被推测性地获取、解码、发布、并执行时,实际的分支结果38最后被从执行阶段18返回,并且这可以与分支预测缓存40-42的相关条目中当前指示的预测进行比较。如果该分支被误预测,则该预测可以被更新以增加预测

在将来正确的可能性。例如,置信度计数器可以递增或递减,并且预测结果可以在不同预测中的置信度升高到足够超过先前预测的情况下被改变。如果实际分支结果38的返回使得预测改变,则这会导致脏标志68被设置,使得这个分支预测状态在稍后时间的逐出将触发写回请求78以更新存储器系统中的相应分支预测状态。

[0067] 当处理在软件工作负载之间切换时,信号80由管线中能够检测工作负载切换的任意元件提供给分支预测器4。例如,获取阶段6可以认识到,指令获取地址已经到达标记不同软件工作负载之间的边界的程序指令地址,或者解码阶段10可以检测到遇到工作负载划分指令并且向分支预测器通知工作负载的切换。另外,执行阶段18可以在检测到工作负载的切换时对该工作负载的切换进行标记并将其用信号通知给分支预测器4。另外,在一些情况下,工作负载的切换可以由中断或异常触发,并且在这种情况下,工作负载切换信号80可以由中断控制器发信号通知。不管工作负载切换信号80的源,响应于工作负载切换信号80,分支预测缓存40-42可以通过清理每个条目的有效位66来使分支预测缓存中的所有条目无效。这确保了任何先前分配的分支预测状态将不被用来进行输入工作负载的预测(当基于以前的工作负载训练时)。如果任何条目64包含由脏标志68标记的脏预测数据,则写回请求78可以由线路占用/写回控制电路44触发以使脏状态被写回到存储器系统中的相应分支预测状态集,使得在输出工作负载随后被再次执行时,其可以得益于基于实际分支结果48训练的任何先前得出的预测。

[0068] 图4示出了工作负载之间的上下文切换的示例。最初工作负载0被执行,然后工作负载切换(或上下文切换)82发生以触发处理管线切换到工作负载1。如上面讨论的,此时,分支预测缓存40-42中的所有条目被无效,并且任何脏状态被写回到与工作负载0相关联的分支预测数据集。为了用与输入工作负载1相关联的分支预测状态填充缓存,工作负载切换82还触发线路占用/写回控制电路44向存储器系统发布针对与工作负载1相关联的分支预测状态集60的至少初始部分的获取请求。在一些情况下,工作负载1的整个分支预测状态集可以在此时被请求。但是,为了改善性能,根据需要以较小的块加载状态可以更高效(例如,每次一个缓存行)。因此,分支预测状态集60可以被分割为由分支预测缓存40-42临时缓存的多个块,但是在需要时可以被写回到存储器层次结构中的较低等级。这减少了上下文切换的延时并且避免了对于总体状态大小的限制。

[0069] 在并非响应于工作负载而加载输入工作负载的所有分支预测状态集的情况下,有可能在缓存当前不具有由获取阶段6提供的目标地址70的有效条目时检测出分支预测缓存40-42中的后续未命中84。因此,如果分支预测缓存中存在未命中,则可以触发另一线路占用请求74来请求工作负载1的分支预测状态集60的相应部分。

[0070] 另一选项可以是提供图1所示的分支预测状态预取器86,其中,分支预测状态预取器86被提供有在分支预测缓存40-42中查找到的指令地址序列,并检测是否检测到连续地址之间存在规则偏置的地址步长模式。当检测到步长模式时,预取器可以触发线路占用/写回控制电路44向存储器系统发布针对与按照预测地址步长的间隔延伸超过当前执行点的地址相关联的分支预测状态数据的附加线路占用请求。当检测到实际地址模式不再遵循步长模式时,可以暂停这样的预取直到检测到另一步长模式为止。例如,用于将指令预取到指令缓存8中的任何已知的预取技术也可以应用于分支预测缓存40、41、42。

[0071] 注意,由于分支预测缓存40-42在上下文切换时被无效,所以输入软件工作负载不

可能命中基于从另一软件工作负载的执行推导得出的结果填充的条目64,这意味着不必利用工作负载标识符来对每个条目64进行标记,这在仍然能够减轻上述攻击的同时减小了实现分支预测缓存40-42的面积开销。

[0072] 分支预测缓存可以缓存分支预测状态,因此可以针对最小粒度大小的相邻地址块进行分支指令结果的独立预测。例如,如果最小粒度大小为例如,16字节,则这可以意味着16字节块中的所有指令将一直具有针对该块预测得出的同一分支结果,但是相邻的16字节块中的指令可以具有相对于先前块进行的不同预测。通过定义一些最小粒度大小,可以减小分支预测器的开销,因为多个分支出现在同一块中通常并不常见,并且可以提供附加结构用于预测这样的多分支情况,或者替代地,BTB中的预测状态72可以被扩充为包括考虑一个块中存在多个分支的情况的信息。

[0073] 但是,尽管可以针对最小粒度大小的相邻块进行独立预测,但是提供充足的分支预测状态项是非常昂贵的(因为完全独立的分支预测项可以针对最小粒度大小的每个可能的地址块被保持在分支预测组60中)。相反,最小粒度大小的不同地址块可以共享同一分支预测状态数据项,从给定指令地址到相应分支预测状态项的映射是基于应用于将较大地址映射到访问该地址的特定分支预测状态项的较小标识符的地址的某种哈希函数确定的。

[0074] 在典型的分支预测器中,这种哈希函数通常被设计为具有尽可能小的局部性。哈希函数的局部性是指地址空间的相邻区域中的地址倾向于被映射到同一分支预测状态项的特性。因此,利用提供非常小的局部性的哈希函数,将有可能在地址空间中的相邻区域中的地址倾向于被映射到不同的分支预测状态项的同时,增加来自地址空间的不同部分的地址共享同一分支预测状态的可能性。期望典型的分支预测器避免程序代码的相邻区域中的分支争用对于同一分支预测条目的访问的热点(这会导致性能降低,因为它们会以冲突方式训练分支预测器,使得相邻区域接收对于它们的实际分支预测结果的适当预测)。相反,地址空间中距离较远的指令地址处的分支不太可能在管线中彼此相邻地被执行,并且在处理到达给定地址之前,针对地址空间中远处的另一地址进行的分支结果的预测不太可能相关。因此,标准的分支预测哈希方案倾向于降低局部性,并且倾向于将来自地址空间的不同部分的地址映射到同一预测状态项(尤其是以分支预测结构本身使用的相同的最小粒度大小的粒度)。

[0075] 相反,利用图1所示的方法(其中,RAM 34被用作所缓存的预测结构40-42的备份存储器),这种传统的分支预测哈希方案将提供较差的性能。如果哈希方案具有非常小的局部性,则这意味着当处理管线通过多个相邻程序指令块时,每个块可以映射到分支预测状态集60中对应于当前软件工作负载的不同分支预测状态项,并且这会导致大量线路占用获取请求被线路占用/写回控制电路44发布从而增大上下文切换延时和传输分支预测状态所需的存储器带宽。因此,标准哈希方案会导致分支预测缓存40-42的性能较差。

[0076] 图5A和5B示出了重新组织用于生成适当的分支预测状态条目的选择以提高局部性的哈希算法和分支预测缓存40的方法。如图5A所示,针对给定软件工作负载的分支预测状态集60可以被划分为某个单位大小的多个子表90,该单位大小使得子表90可以在一次数据传输中被传输到分支预测缓存40-42。即,每个子表90的大小可以对应于一个缓存行。每个子表可以包括多个单独的预测状态项,每个预测状态项对应于最小粒度大小S的给定地址范围。因此,大小为S的每个地址范围可以具有分别根据针对大小为S的相邻地址块所做

的预测进行的独立预测。

[0077] 如图5的左手侧大小所示,用来选择用于进行给定地址的预测的适当子表90的哈希函数可以使得地址被映射到对应于比大小S更大的大小T的连续地址范围的块中的子表。因此,在本示例中,图5A中被标记为B的连续地址范围中的所有地址可以分别映射到相同的子表90-B。如图5A的上部所示,这可以例如,通过使用目标地址的第一部分94来选择将访问的特定子表90,然后将用于选择该子表中的单独预测条目的哈希函数96应用于该地址的剩余(第二)部分98来实现。在本示例中,第一部分94在地址的中间部分,因此第二部分98包括比第一部分更低的有效位和比第一部分更高的有效位。在其他示例中,第一部分94可以在地址的上端,因此只有较低有效位被输入到哈希函数96。一些地址位可以不被用于子表选择或哈希函数96(例如,如图5A所示,较低部分可以被忽略)。应用于地址的第二部分的哈希函数96可以为与被设计为提供非常小的局部性的标准分支预测哈希函数相类似的设计。但是,从该哈希函数中排除第一部分94,意味着相比这种标准哈希函数局部性增加。这意味着对于大小为T的连续地址范围中大小为S的所有地址块,预测大小为T的地址范围中所有分支的分支结果所需要的所有相关分支预测状态可以在一次数据传输中被传输到分支预测缓存40-42中。这有助于通过减少预测大小为T的范围中的地址的分支结果所需要的对于存储器的单独访问的次数来提高性能。

[0078] 在一些示例中,除了目标地址外,BHB 41中包括的分支历史信息也可以被用作用于生成对分支预测状态的索引的哈希函数96的附加输入。这可以有助于区分去往程序中的同一分支指令的不同路线(取决于先前分支的结果),从而有助于进行更精确的预测。

[0079] 图5B示出了当如图5A所示布置分支预测状态时使用的分支预测缓存的替代布置。另外,在图5B中,每个条目64都具有图3所示的有效位66、脏位68、以及地址标志70。但是,图5中的预测字段72缓存基于图5A所示的目标地址的第一部分94而选择的子表90。如条目64之一所示,每个子表随后可以包括分别对应于最小粒度大小为S的不同地址块的多个单独的预测状态项100。因此,利用这种布置,每个条目64对应于大小为T的地址块,并且包含针对大小为S的不同块的单独预测,其中,T大于S。在这种情况下,当访问分支预测缓存时,选择访问哪个条目64的索引102是按照图5A所示的相同方式从目标地址的第一部分94得出的,然后地址的第二部分98(可选地,来自BHB 41的分支历史)被输入哈希函数96,该哈希函数随后被用来选择图5B的下部所示的索引条目64中的特定预测项100。

[0080] 因此,在图5A和5B中,每个缓存条目充当针对地址空间的某个子集的独立BP/BTB。例如,如果选择大小为64字节的缓存颗粒并且可用的RAM空间为16k字节,则我们可以将256个独立BP/BTB存储在RAM中。执行查找时的简单方案将使用源地址的19:12位来判定在哪个BP/BTB中执行查找,然后按照惯常做法在64字节的子BTB/BP中执行BP/BTB查找。该特定示例会导致每兆字节的混淆,并且可以使用哈希来缓解该混淆。

[0081] 图6示出了示出在分支预测器4中查找针对给定目标指令地址的预测的方法的流程图。在步骤150,目标指令地址被供给到分支预测器4。在步骤152,与分支预测缓存40-42相关联的缓存查找电路确定是否存在针对所查找地址的命中或未命中。当缓存结构的索引条目包括有效条目时发生命中,其中,该有效条目的地址标志70匹配目标指令地址的相应部分。一般地,参考分支预测缓存描述了图6,但是将理解的是,这可以是方向预测缓存40或分支目标缓冲器缓存42。

[0082] 如果分支预测缓存中存在命中,则在步骤154基于命中目标地址的匹配条目来确定分支指令的预测结果。预测结果可以被返回给获取阶段6,然后获取阶段6可以基于预测出的分支结果获取后续指令。一旦对应于目标指令地址的指令的实际结果在执行阶段18已知,则其可以被用信号发送回分支预测器,并且在步骤156,分支预测器可以将实际结果与所指示的预测状态进行比较,并在需要时更新预测状态以改变预测置信度等级或预测本身。

[0083] 如果在步骤152在分支预测缓存40-42中检测到未命中,则在步骤160线路占用和写回控制电路44触发请求将对应于当前工作负载的预测状态数据集60的相应部分获取到相关分支预测缓存40-42中的状态获取请求(或线路占用请求)。此时,条目选择电路45可以通过应用图5A所示的方法选择适当的子表然后将其映射到存储器中对应于该子表的相关地址,来生成相关分支预测状态的地址。例如,线路占用和写回控制电路中的条目选择逻辑45可以具有指定针对各个工作负载的每个子表的地址的表。在步骤162,在分支预测缓存40-42中分配用于在预测状态的相应部分响应于线路占用请求被返回时对其进行存储的条目。如果选择已经包含脏数据的受害者条目,则该脏预测状态被写回到存储器系统。在分支预测缓存40-42中与输出工作负载有关的所有条目响应于上下文切换被写回的系统中,可以认为由分支预测缓存中的未命中触发的任何写回与当前工作负载有关,并且在这种情况下不需要跟踪分支预测器处超出上下文切换的先前的工作负载的任何标识。在替代地通过将写回推迟到需要后续覆写为止来减少上下文切换的开销的系统中,分支预测缓存可以存储先前执行的工作负载的附加标识符,并且可以在每个条目64中包括标记具有脏数据的无效条目与当前工作负载还是先前执行的工作负载有关的标志,因此可以确定脏数据需要被回写到的分支预测状态集的写回。

[0084] 由于线路占用请求被存储器系统处理会花费一些时间,所以在步骤164,分支预测器4可以返回默认分支预测,作为针对目标指令地址的预测结果72。例如,默认预测可以是在不存在可用的分支预测状态的情况下应该预测不采用分支。通过进行默认预测,使得在等待相关预测状态被返回的同时后续指令继续被获取,从而减少了处理后续指令的延迟。如果默认预测证明是不正确的,则可以按照类似于其他分支误预测的方式进行处理。在步骤166,当分支预测状态响应于状态获取请求被从存储器系统30、32、34返回时,其被写入分支预测缓存40-42的在步骤162分配的条目。如果实际分支结果38此时已经可用并且其与所获取的状态数据指示的当前值不一致,则在必要时也可以按照如以上的步骤156所述的对缓存的现有条目进行更新的类似方式基于分支结果对所获取的状态数据进行更新。

[0085] 图7示出了在分支预测器4处理工作负载的切换的方法。在步骤170,基于工作负载切换信号80检测处理工作负载的切换。如上面讨论的,该信号可以由例如中断控制器、执行阶段80、或获取或解码阶段6、10提供。在步骤172,响应于工作负载的切换,分支预测缓存40-42更新其所有条目,以通过清理有效标志66将每个条目64标记为无效。这阻止了输入工作负载命中由先前工作负载分配的条目。在步骤174,通过发布写回请求78,任何脏预测状态(在脏标志68被设置的条目中指示)被写回到存储器。替代地,写回可以被推迟到后续线路占用获取76需要覆写该条目为止,但是在这种情况下,需要对于哪些条目与输入工作负载有关以及哪些条目与输出工作负载有关的一些附加追踪。注意,这种追踪仍然可以比利用完整的负载标识符对每个条目进行标记更高效,因为单个位标志将足以在每个缓存条目

64中区分输入和输出工作负载。在所有条目之间共享的输出工作负载的标识符的单独存储可以足够识别输出工作负载,而无需在每个缓存条目64中单独对其进行识别。替代地,其他微架构实施方式可以偏好通过响应于上下文切换简单地发布所有写回请求来简化写回处理,因此一旦当前工作负载的处理开始即无需考虑来自先前执行的工作负载的任何剩余脏数据。

[0086] 在步骤176,响应于软件工作负载的切换,线路占用/写回控制电路44还生成请求获取针对新输入工作负载的分支预测状态集60的至少一部分的一个或多个状态获取请求74。响应于工作负载的切换请求的该部分可以是完整的分支预测状态集60或者可以是更小部分,剩余部分在必要时根据需要被获取,以产生对于在新工作负载的后续处理中遇到的目标地址70的预测。

[0087] 在本申请中,措辞“被配置为...”用来表示装置的元件具有能够实现所限定的操作的配置。在本上下文中,“配置”是指硬件或软件的互连方式或部署。例如,该装置可以具有提供所限定的操作的专用硬件,或者处理器或其他处理设备可以被编程为执行该功能。“被配置为”并不意味着该装置元件需要以任何方式被改变以提供所限定的操作。

[0088] 尽管本文参考附图描述了本发明的说明性实施例,但是将理解的是,本发明不限于这些精确实施例,本领域技术人员可以在不偏离所附权利要求限定的本发明的范围和精神的条件下做出各种修改和改变。

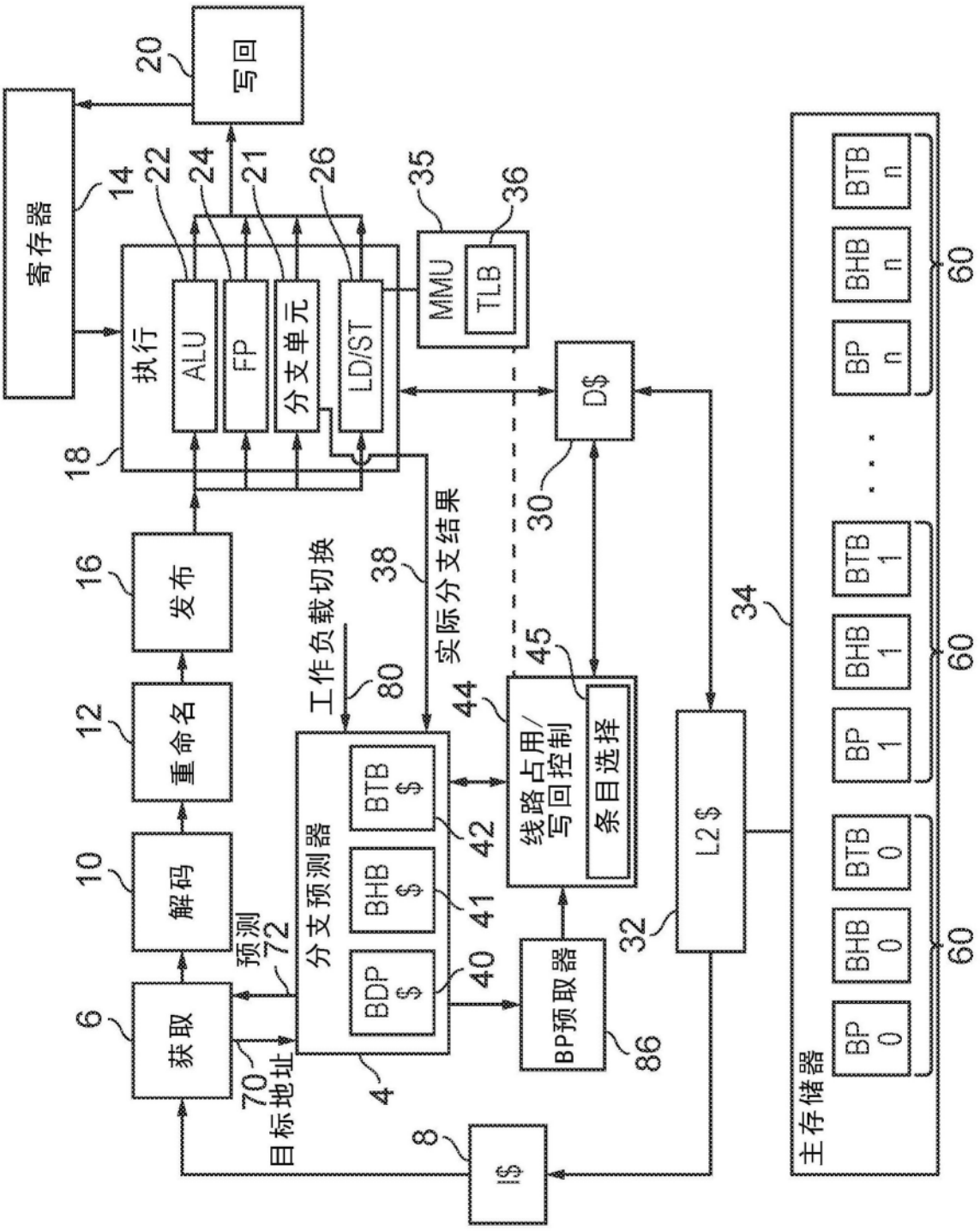

寄存器
14
写回
20
执行
ALU
FP
分支单元
LD/ST
22
24
21
26
MMU
TLB
35
36
18
发布
16
重命名
12
解码
10
获取
6
预测
72
目标地址
70
工作负载切换
80
分支预测器
BDP $
BHB $
BTB $
40
41
42
4
实际分支结果
38
44
线路占用/写回控制
条目选择
45
BP预取器
86
D$
30
L2 $
32
34
I$
8
主存储器
BP 0
BHB 0
BTB 0
BP 1
BHB 1
BTB 1
BP n
BHB n
BTB n
60

图1

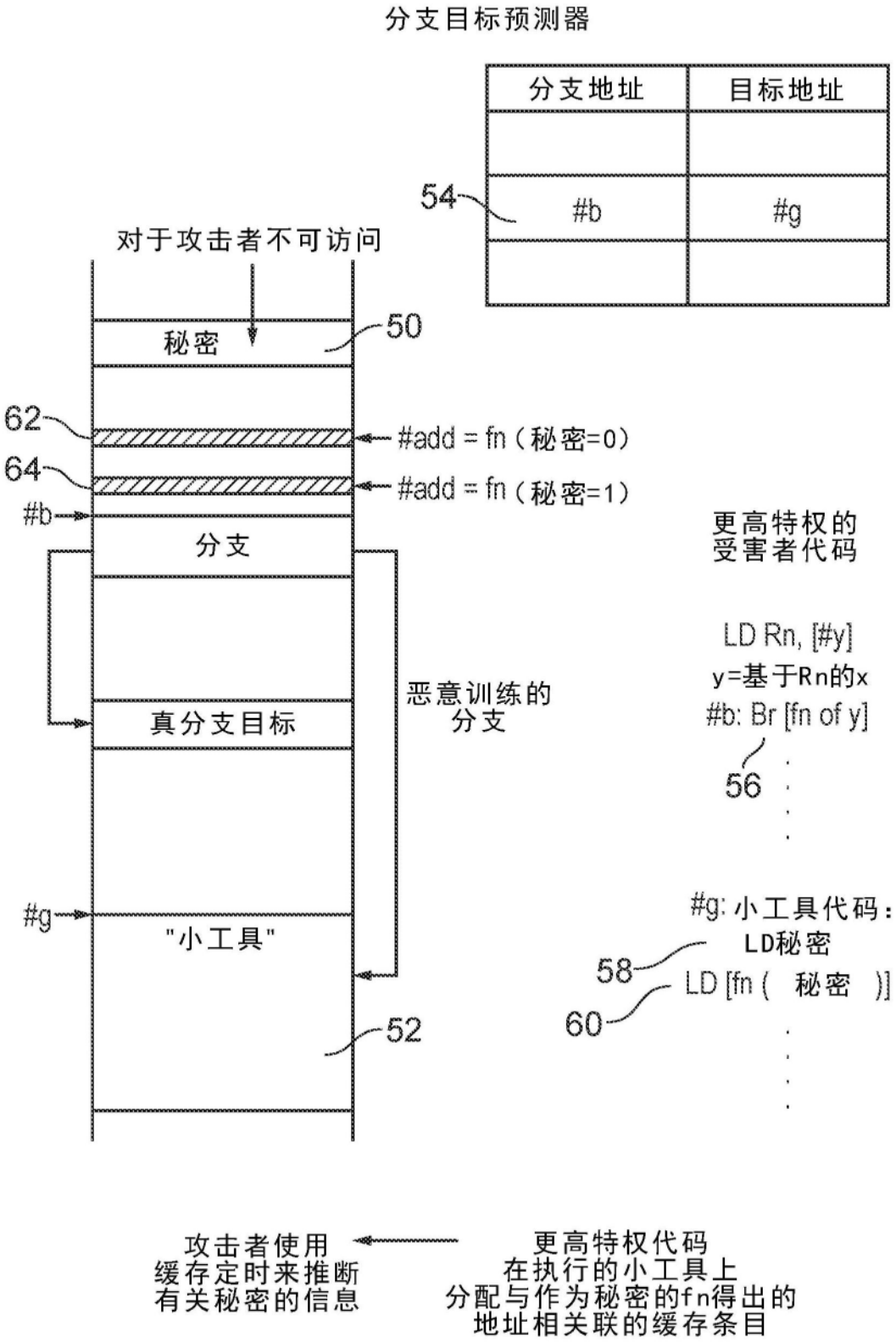

分支目标预测器
分支地址
目标地址
#b
#g
54
对于攻击者不可访问
秘密
50
62
#add = fn（秘密=0）
64
#add = fn（秘密=1）
#b
分支
更高特权的
受害者代码
LD Rn, [#y]
y=基于Rn的x
#b: Br [fn of y]
56
恶意训练的
分支
真分支目标
#g
"小工具"
#g: 小工具代码：
LD秘密
58
LD [fn (秘密)]
60
52
攻击者使用
缓存定时来推断
有关秘密的信息
更高特权代码
在执行的小工具上
分配与作为秘密的fn得出的
地址相关联的缓存条目

图2

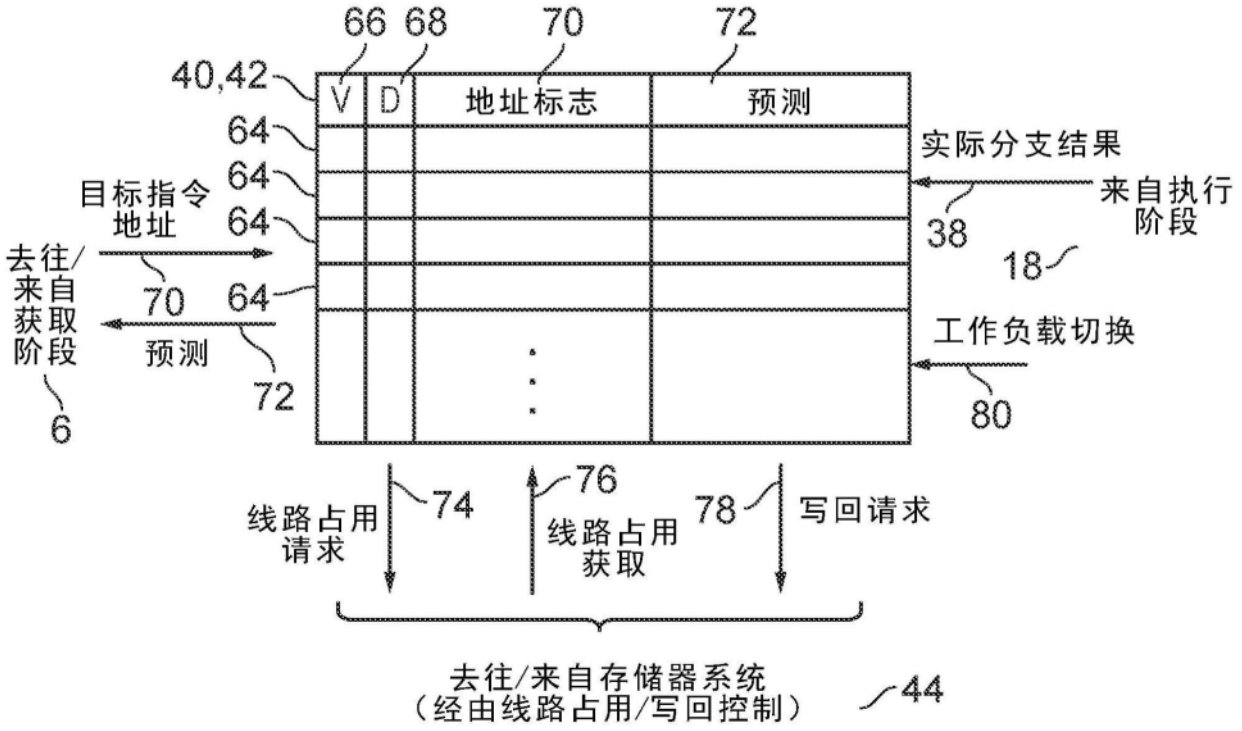

66
68
70
72
40,42
V
D
地址标志
预测
64
64
64
64
实际分支结果
来自执行阶段
38
18
目标指令地址
去往/来自取获阶段
70
预测
72
6
工作负载切换
80
线路占用请求
74
76
线路占用获取
78
写回请求
去往/来自存储器系统（经由线路占用/写回控制）
44

图3

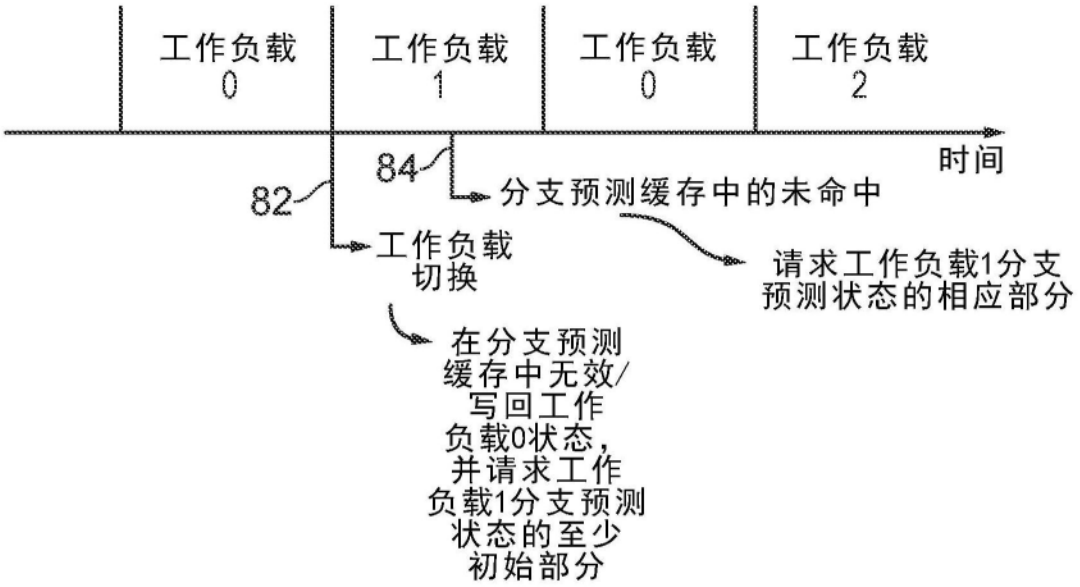

工作负载 0
工作负载 1
工作负载 0
工作负载 2
时间
84
分支预测缓存中的未命中
82
工作负载切换
请求工作负载1分支预测状态的相应部分
在分支预测缓存中无效/写回工作负载0状态，并请求工作负载1分支预测状态的至少初始部分

图4

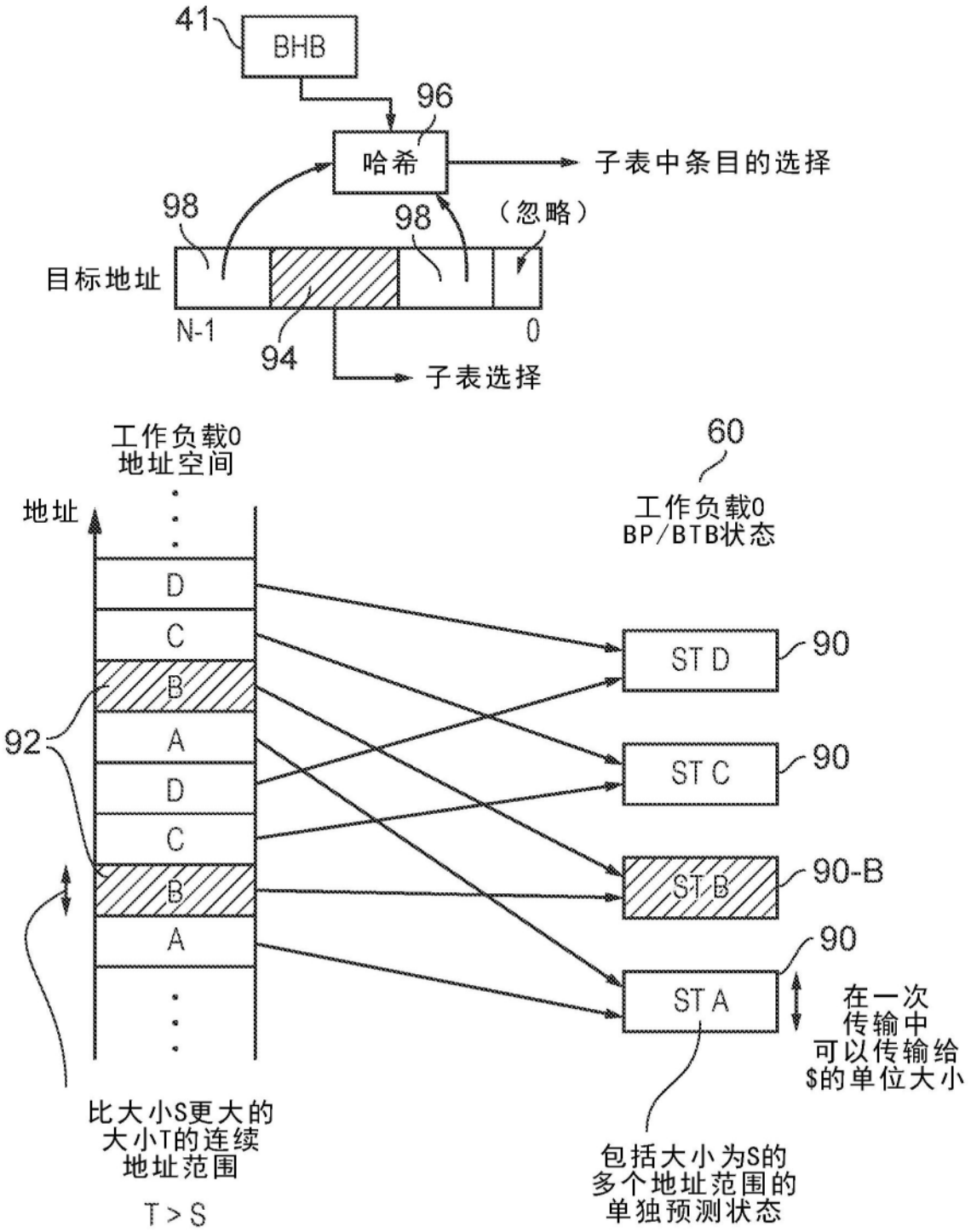

41
BHB
96
哈希
子表中条目的选择
98
98
（忽略）
目标地址
N-1
0
94
子表选择
工作负载0
地址空间
地址
60
工作负载0
BP/BTB状态
D
C
B
A
D
C
B
A
92
ST D
90
ST C
90
ST B
90-B
ST A
90
在一次
传输中
可以传输给
$的单位大小
比大小S更大的
大小T的连续
地址范围
T > S
包括大小为S的
多个地址范围的
单独预测状态

图5A

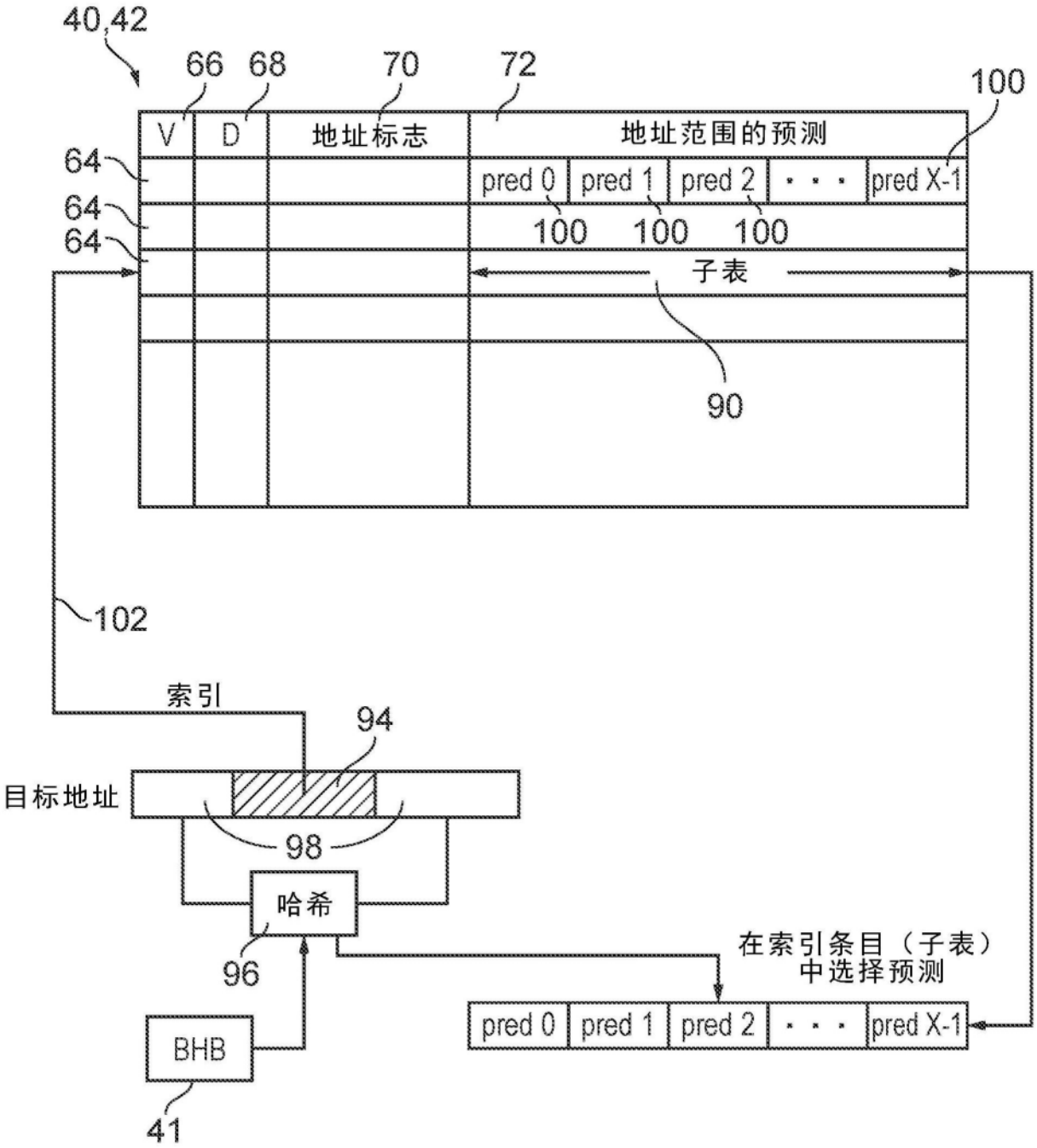

40,42
66
68
70
72
100
V
D
地址标志
地址范围的预测
64
64
64
pred 0
pred 1
pred 2
· · ·
pred X-1
100
100
100
子表
90
102
索引
94
目标地址
98
哈希
96
BHB
41
在索引条目（子表）
中选择预测
pred 0
pred 1
pred 2
· · ·
pred X-1

图5B

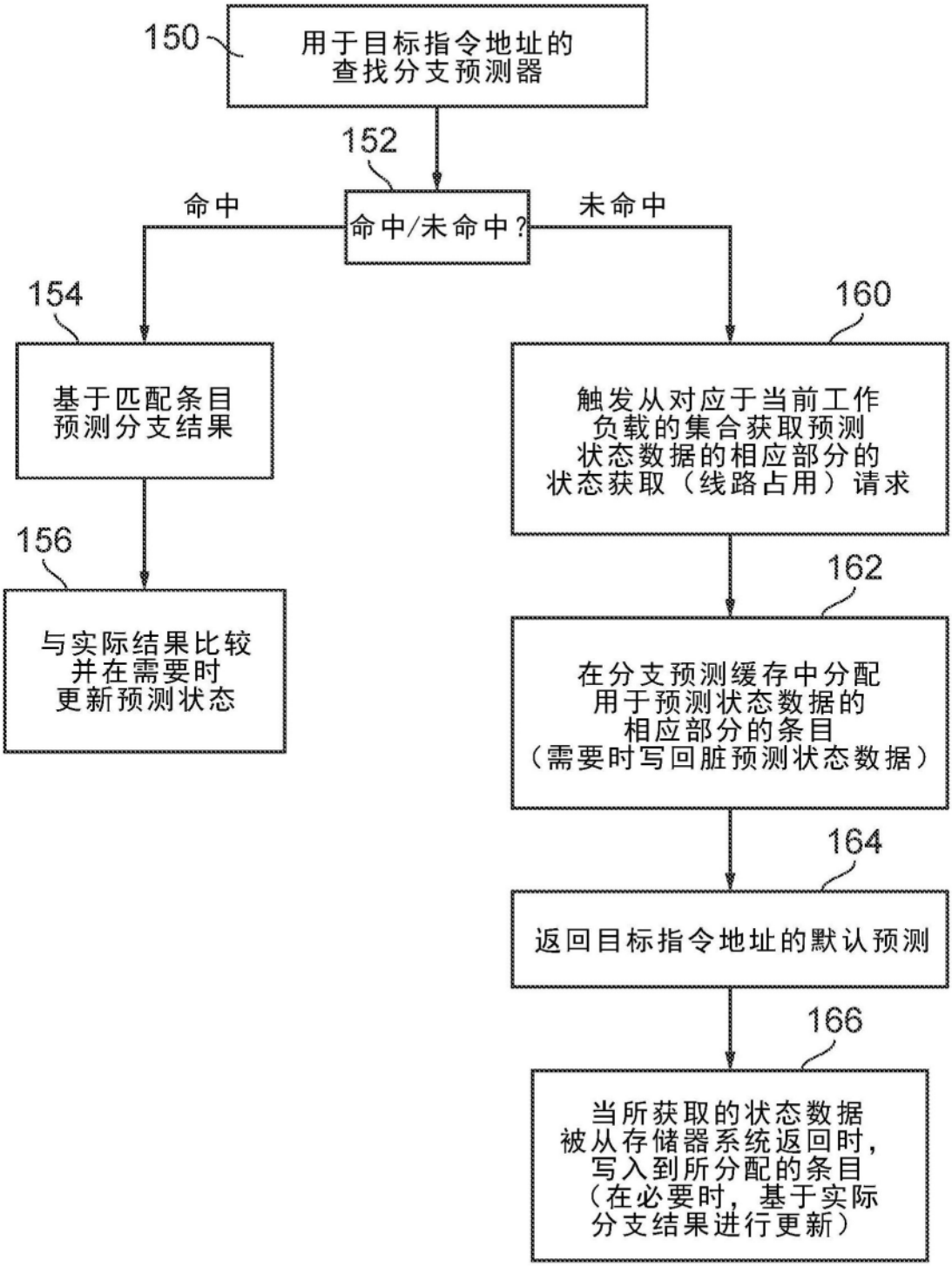

150
用于目标指令地址的查找分支预测器
152
命中/未命中？
命中
未命中
154
基于匹配条目预测分支结果
156
与实际结果比较并在需要时更新预测状态
160
触发从对应于当前工作负载的集合获取预测状态数据的相应部分的状态获取（线路占用）请求
162
在分支预测缓存中分配用于预测状态数据的相应部分的条目（需要时写回脏预测状态数据）
164
返回目标指令地址的默认预测
166
当所获取的状态数据被从存储器系统返回时，写入到所分配的条目（在必要时，基于实际分支结果进行更新）

图6

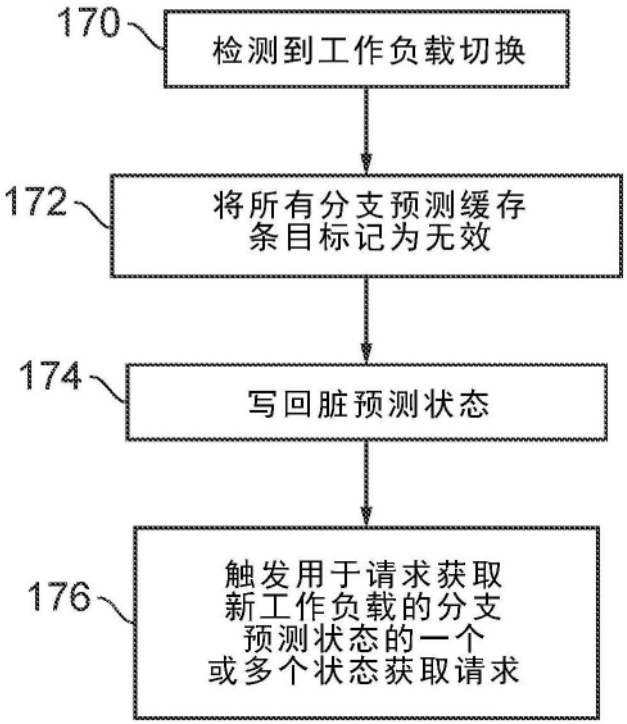

170
检测到工作负载切换
172
将所有分支预测缓存
条目标记为无效
174
写回脏预测状态
176
触发用于请求获取
新工作负载的分支
预测状态的一个
或多个状态获取请求

图7