



(10)授权公告号 CN 106919367 B

(21)申请号 201710138752.8

(51) Int.Cl.

(22)申请日 2017.03.09

G06F 9/30(2006.01)

(65)同一申请的已公布的文献号

(56)对比文件

申请公布号 CN 106919367 A

US 2015324213 A1, 2015.11.12,

(43)申请公布日 2017.07.04

US 2012296877 A1, 2012.11.22,

(30) 优先权数据

US 6009516 A, 1999.12.28,

62/324,945 2016.04.20 US

CN 101894010 A, 2010.11.24,

15/156.416 2016.05.17 US

CN 104615548 A, 2015.05.13.

CN 1521635 A, 2004.08.18,

(73)专利权人 上海兆芯集成电路有限公司

宙查员 康健

地址 201203 上海市浦东新区上海市张江

高科技园区金科路2537号301室

(72)发明人 布兰特·比恩 柯林·艾迪

(74)专利代理机构 北京林达刘知识产权代理事

务所(普通合伙) 11277

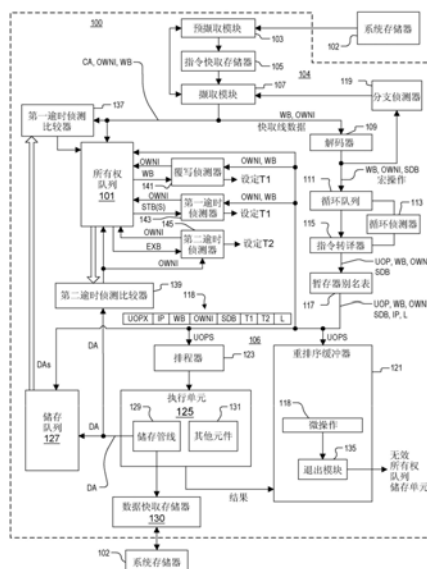
代理人 刘新宇

权利要求书4页 说明书15页 附图5页

侦测自修正程序码的处理器与方法

(57)摘要

一种侦测自修正程序码的处理器与方法,该处理器与方法依据快取线决定存储器所有权以侦测自修正程序码。所有权队列储存有快取线地址与对应的所有权索引。快取线数据被转译成指令,且每一指令与相关联的储存单元的所有权索引一同被提供。新的每一快取线地址被相对于每一储存指令的目标地址进行比较,且当决定时,每一目标地址被比较于该所有权队列中的每一快取线地址。匹配的储存单元被标记为逾时的,且当依据逾时的储存单元所产生的每一指令准备好退出时产生例外事件。借此,快取线与对应的储存指令间的配对产生例外事件。例外事件刷新处理器以解决潜在的修正的程序码情况。本发明能够提高处理器的效能。



1. 一种处理器,其特征在于,用于依据快取线决定存储器所有权以侦测自修正程序码,该处理器包括:

所有权队列,包括多个储存单元;

撷取系统,用以提供多条快取线的快取线数据给处理系统的处理前端,其中,针对每一该快取线,该撷取系统用以决定所有权索引,并且将该所有权索引与对应的快取线地址输入进该所有权队列的该多个储存单元的其中之一;

其中,该处理前端用以将该多条快取线的该快取线数据转换成多个指令,每一该指令包括该所有权队列中储存有快取线地址的储存单元的所有权索引,该快取线地址对应于产生每一该指令的该快取线数据,且该处理前端用以发出每一该指令以执行该指令;

其中,该处理系统还包含执行系统,该执行系统用以决定被发出的每一储存指令的目标地址,当该所有权队列中具有与将要退出的指令相匹配的所有权索引的储存单元的逾时位被设定时,该执行系统执行第一例外事件;

第一比较器,用以将被输入进该所有权队列的每一该快取线地址与已被决定的每一该目标地址进行比较,当决定该多条快取线地址的其中之一匹配于所述目标地址的其中之一时,该第一比较器用以设定该逾时位;以及

第二比较器,当每一该目标地址被该执行系统决定时,该第二比较器用以将该目标地址与该所有权队列中储存的有效储存单元的每一快取线地址进行比较,且该第二比较器还用以设定每一匹配的储存单元的该逾时位,

其中,当输入该快取线地址时,该撷取系统令该所有权队列中的对应的储存单元为有效储存单元。

2. 根据权利要求1所述的处理器,其特征在于,该第一例外事件使该执行系统刷新该处理器,以避免产生该第一例外事件的指令退出,且使该撷取系统自指令快取存储器中重新撷取产生该第一例外事件的指令。

3. 根据权利要求1所述的处理器,其特征在于,该处理前端将该所有权队列中的该多个储存单元中的对应的一个储存单元的最后指令标记为最后指令;

并且,当退出的指令被标记为该最后指令时,该执行系统使该所有权队列中的该多个储存单元中的该对应的一个储存单元无效。

4. 根据权利要求1所述的处理器,其特征在于,还包含:

逾时侦测器,用以使用被发出的指令的该所有权索引读取该所有权队列的对应的储存单元的逾时位,且当该对应的储存单元的该逾时位被设定时,该逾时侦测器用以令被发出的该指令被标记,以调用该第一例外事件;

其中,当将要退出的指令被标记以调用该第一例外事件时,该执行系统执行该第一例外事件。

5. 根据权利要求4所述的处理器,其特征在于,

该处理前端还用以设定每一指令中的跨立位,该指令源自跨立两条快取线的快取线数据;以及

其中,当被发出的该指令所具有的跨立位被设定时,该逾时侦测器用以读出该所有权队列中的该对应的储存单元的连续的下一个储存单元的该逾时位,并且,当该所有权队列中的该对应的储存单元的连续的下一个储存单元的该逾时位被设定时,该逾时侦测器令被

发出的该指令被标记以调用该第一例外事件。

6. 根据权利要求1所述的处理器,其特征在于,该撷取系统用以决定该所有权索引为二进制计数值,该二进制计数值随着每一储存单元被输入进该所有权队列而增加,该二进制计数值的总计数量至少是该所有权队列中的储存单元的总数量;

其中,该所有权索引的最高有效位包含缠绕位;

该处理器还包含覆写侦测器,该覆写侦测器用以使用被发出的指令的该所有权索引读取该所有权队列中的对应的储存单元的缠绕位,且当该对应的储存单元的该缠绕位不匹配该被发出的指令对应的该缠绕位时,该覆写侦测器用以令该被发出的指令被标记以调用该第一例外事件;

其中,当该第一例外事件被标记于将要退出的指令时,该执行系统执行该第一例外事件。

7. 根据权利要求1所述的处理器,其特征在于,还包括:

储存队列,包括多个储存单元,每一该储存单元用以储存被该处理前端发出的储存指令,且每一该储存单元用以储存该执行系统决定的目标地址;

该执行系统还包括储存管线,该储存管线用以决定被调度以执行的每一储存指令的目标地址,且用以将决定出的每一目标地址提供至该储存队列中的对应的储存单元和该第二比较器。

8. 根据权利要求1所述的处理器,其特征在于,

该处理系统使用被发出的指令的该所有权索引存取该所有权队列中的对应的储存单元,以设定该对应的储存单元中的执行位;

该处理器还包含:逾时侦测器,用以计算被第二比较器决定的每一该匹配的储存单元的该执行位,且当任何匹配的储存单元的任何执行位被设定时,该逾时侦测器还用以使对应于被决定的该目标地址的储存指令被标记以调用第二例外事件;

其中,当将要退出的储存指令被标记以调用该第二例外事件时,该执行系统执行该第二例外事件。

9. 根据权利要求8所述的处理器,其特征在于,该第二例外事件使该执行系统允许被标记以调用该第二例外事件的该储存指令退出,刷新该处理器,且使该撷取系统取得指令指针以从指令快取存储器读取该储存指令后的指令。

10. 根据权利要求1所述的处理器,其特征在于,该处理前端还用以设定源自跨立于两快取线的快取线数据的每一指令的跨立位;

其中,该处理系统使用被发出的指令的该所有权索引存取该所有权队列中的对应的储存单元,以设定该对应的储存单元的执行位,且当该被发出的指令的该跨立位被设定时,设定在该对应的储存单元之后的下一个连续的储存单元的执行位。

11. 根据权利要求10所述的处理器,其特征在于,还包含:

逾时侦测器,用以计算被第二比较器决定的每一该匹配的储存单元的该执行位,且当任何匹配的储存单元的任何执行位被设定时,该逾时侦测器还用以使对应于被决定的该目标地址的储存指令被标记以调用待执行的第二例外事件;

其中,当将要退出的储存指令被标记以调用该第二例外事件时,该执行系统执行该第二例外事件,其中,该第二例外事件使该执行系统允许被标记以调用该第二例外事件的该

储存指令退出,刷新该处理器,且使该撷取系统取得指令指针以从指令快取存储器读取该储存指令之后的指令。

12.一种依据快取线决定存储器所有权以侦测自修正程序码的方法,其特征在于,包括:

取得多条快取线,每一该快取线具有快取线地址与快取线数据,针对取得的每一该快取线,决定所有权索引,并将该快取线地址与该所有权索引输入进所有权队列中的多个储存单元的其中之一;

当每一该快取线地址被输入进该所有权队列的储存单元时,比较该快取线地址与依据被发出的每一储存指令而被决定的每一目标地址,且当快取线地址其中之一与目标地址其中之一相匹配时,将匹配的储存单元标记为逾时的;

转换该多条快取线的快取线数据为多个指令,每一指令包括为该所有权队列中的储存单元而决定的所有权索引,该储存单元储存有快取线,该指令转换自该快取线;

发出多个指令以执行;

决定被发出以执行的每一储存指令的目标地址;

当每一目标地址被决定时,比较被决定的该目标地址与该所有权队列中每一有效储存单元的该快取线地址,并标记任何匹配的储存单元为逾时,其中,当接收到新的快取线地址时,使该所有权队列中的对应的储存单元为有效储存单元;以及

当将要退出的指令的所有权索引匹配于该所有权队列中被标记为逾时的储存单元时,执行第一例外事件。

13.根据权利要求12所述的方法,其特征在于,该执行第一例外事件的步骤包含刷新处理器,避免用以调用该第一例外事件的该指令退出,以及重新撷取调用该第一例外事件的指令。

14.根据权利要求12所述的方法,其特征在于,还包含:

标记该所有权队列中的每一有效储存单元的最后指令为最后指令;以及当被标记为最后指令的指令已经退出时,使该所有权队列中对应的储存单元无效。

15.根据权利要求12所述的方法,其特征在于,还包含:

当每一该指令被发出时,通过被包含的所有权索引存取该所有权队列中对应的储存单元,且当该对应的储存单元被标记为逾时的时候,标记该指令以调用该第一例外事件;

其中,该执行第一例外事件的步骤还包含:为将要退出且被标记以调用该第一例外事件的每一指令执行该第一例外事件。

16.根据权利要求12所述的方法,其特征在于,还包含:

在该转换的过程中,对从跨立该多条快取线中的两条快取线的快取线数据转换成的每一指令设定跨立位;

当每一指令被发出时,使用每一该指令包括的该所有权索引存取该所有权队列中对应的储存单元,且当该跨立位被设定时,存取该所有权队列中下一个连续的储存单元;

当该对应的储存单元被标注为逾时的时候,标记该指令以调用该第一例外事件;以及

当该跨立位被设定且该所有权队列中的下一个连续的储存单元被标记为逾时的时候,标记该指令以调用该第一例外事件;

其中,该执行第一例外事件的步骤包含:为将要退出且被标记以调用该第一例外事件

的每一指令执行该第一例外事件。

17. 根据权利要求12所述的方法, 其特征在于, 还包含:

重复地增加二进制计数值的该所有权索引, 该二进制计数值的总计数量至少是该所有权队列中的储存单元的总数量;

决定该二进制计数值的最高有效位为缠绕位;

其中, 该转换的步骤包含: 使对应的缠绕位包含于被转换的每一该指令, 被转换的每一该指令包含的该缠绕位与该所有权队列的储存有被转换的指令所转换自的快取线的储存单元中被决定的缠绕位相同;

使用指令中的所有权索引存取该所有权队列中的储存单元, 并比较指令中的缠绕位与所存取的该储存单元的缠绕位;

当所述缠绕位并不匹配时, 标记该指令以调用该第一例外事件; 以及

其中, 该执行第一例外事件的步骤包括: 为被标记以调用该第一例外事件的每一指令执行该第一例外事件。

18. 根据权利要求12所述的方法, 其特征在于, 还包含:

当发出指令时, 使用该指令中的所有权索引存取该所有权队列中的该储存单元, 且设定所存取的该储存单元的执行位;

当匹配的储存单元于比较被决定的目标地址与该所有权队列中每一有效储存单元的快取线地址的过程中被找到时, 决定该匹配的储存单元中的执行位是否被设定;

且当该匹配的储存单元的该执行位被设定时, 标记对应于被决定的目标地址的储存指令以调用第二例外事件。

19. 根据权利要求18所述的方法, 其特征在于, 还包含: 当将要退出的储存指令被标记以调用该第二例外事件时, 允许该储存指令退出与结束, 刷新处理器, 且通过取得在程序顺序中该储存指令后的下一个指令重新开始操作。

20. 根据权利要求12所述的方法, 其特征在于, 还包含:

于该转换的步骤中, 对源自跨立该多条快取线中的两条快取线的快取线数据的每一指令设定跨立位;

当发出指令时, 使用该指令具有的所有权索引存取该所有权队列中的储存单元, 并设定所存取的该储存单元中的执行位, 且当该指令的该跨立位被设定时, 还存取该所有权队列中下一个连续的储存单元, 并设定所述的下一个连续的储存单元的执行位;

当匹配的储存单元于比较被决定的该目标地址与该所有权队列的每一有效储存单元的快取线地址的过程中被找到时, 决定该匹配的储存单元的执行位是否被设定;

并且当该匹配的储存单元的该执行位被设定时, 标记对应于被决定的该目标地址的储存指令以调用第二例外事件。

21. 根据权利要求20所述的方法, 其特征在于, 还包含:

当将要退出的储存指令被标记以调用该第二例外事件时, 允许该储存指令退出并结束, 刷新处理器, 且通过取得在程序顺序中该储存指令后的下一个指令重新开始操作。

侦测自修正程序码的处理器与方法

技术领域

[0001] 本发明关联于存储器所有权,特别是关联于基于快取线决定存储器所有权以侦测自修正程序码。

背景技术

[0002] 自修正程序码(self modified code,SMC)具有被本地处理器执行的至少一指令以修正另一指令,或者是后续被处理器处理的一程序码序列。自修正程序码可能具有一序列的程序码以修正刚被执行的程序码,使得被修正而具有新功能的程序码再被执行一次。于另一个例子中,自修正程序码用以修正顺序紧接的程序码且就在之前被执行。虽然自修正程序码现在并不如往昔普遍,许多旧式的程序仍具有自修正程序码且应被适当的执行。处理器必须能够侦测自修正程序码且校正运算以避免不适当的结果。此处所用的“处理器”一词用以代表任意类型的处理单元,包含微处理器(micro processor)、中央处理器(central processing unit,CPU)、一运算核或一微控制器(micro controller)等等。此处所用的“处理器”一词还包含了任意类型的处理器架构,例如整合有多个处理单元的晶片,或者是包含了具有一系统晶片(system of a chip,SOC)的集成电路(integrated circuit,IC)。

[0003] 现代的处理器的经常执行预撷取运算以将存储器中一或多行读进指令快取存储器(icache)。指令快取存储器的快取线被解析为指令并被执行。为了使效率最大化,撷取单元或者是相仿的元件会试图填满指令快取存储器并维持填满的状态以确保指令被连续地供给执行。为了使效能最大化,执行管线(execution pipeline)被希望能够维持满载的状态。现代的处理器的经常采用乱序(out-of-order,OOO)执行,也就是说晚收到但已准备好被执行的指令可以先于早收到但还没准备好被执行指令被执行。预撷取与乱序运算的至少一个问题是,被预撷取与提供执行的指令可能会在之后被自修正程序码修改。因此,已经被提供执行的指令可能会错过修正,而可能导致不适当或非原意的运算。

[0004] 现代的处理器的需要侦测或预防逾时指令被完成,逾时指令是指被程序码修改之后不欲被执行的指令。处理器一般会将存储器的所有权分割为一指令区域与一数据区域,指令快取存储器拥有指令区域,数据快取存储器(data cache,dcache)拥有数据区域。指令区域被预设为只储存用以执行的指令,而数据区域被预设为可被软件程序所储存的数据与信息所利用。如果指令快取存储器尝试读取数据快取存储器所拥有的存储器,所有权必需被从数据快取存储器转换,且转换的过程会缓慢而冗长且使得运算被序列化。

[0005] 在以往的架构中,所有权基于分页的边界。通常一个分页的大小是4KB(kilobytes)。虽然,4KB的存储器并不占有显着的容量,自修正程序码可能会产生指令快取存储器与数据快取存储器之间的所有权颠簸(thrashing)现象,而降低了运算效率。一种解决方法是降低所有权的大小至四分之一页,也就是4KB的分页大小中的1KB的存储器区块。但尽管只是1KB的所有权区块在许多情况中还是足以对自修正程序码造成麻烦。而且,更大的分页大小也常被使用,像是2MB(megabytes)甚至是1GB(gigabyte),因此对于降低整体效能

来说,所有权区块一直是个重要的议题。

发明内容

[0006] 依据一实施例,一种处理器用以基于快取线决定存储器所有权以侦测自修正程序码,处理器具有所有权队列、撷取系统、处理系统与比较器。处理系统具有处理前端与执行系统。所有权队列储存有储存单元,每一储存单元对应于其中一快取线。撷取系统将快取线的快取线数据提供至处理前端,以决定每一快取线的所有权索引,且将所有权索引与对应的快取线地址输入至所有权队列的储存单元。处理前端将快取线数据转译为指令并提出每一指令以进行执行。每个指令具有所有权队列中对应的储存单元的所有权索引。执行系统决定每一储存指令的目标地址,且当所有权队列的储存单元的逾时位被设定时,执行第一例外事件。所述的储存单元具有匹配的所有权索引与已准备好被退出的指令。第一比较器比较被输入进所有权队列的每一快取线地址与每一目标地址,并在一个相匹配的结果被找到时设定对应的逾时位。第二比较器比较被执行系统决定的每一目标地址与存在所有权队列中的每一快取线地址,并设定每匹配的一储存单元的逾时位。

[0007] 第一例外事件被执行时会使执行系统刷新处理器,以避免触发第一例外事件的指令退出,且第一例外事件还用以使撷取系统重新撷取指令。当输入对应的快取线地址时,撷取系统会使所有权队列中的储存单元有效。处理前端会标注所有权队列中的储存最后指令为最后指令。在此实施例中,当对应快取线的退出的指令被标记为最后指令时,执行系统会使储存单元无效化。

[0008] 在一实施例中,处理器会具有一逾时侦测器。逾时侦测器依据被提出的指令中的所有权索引读取所有权快取存储器中对应的储存单元中的逾时位。且当对应的储存单元中的逾时位被设定时,逾时侦测器用以使被提出的指令被标记以产生第一例外事件。在此实施例中,当准备好退出的指令被标记以产生第一例外事件时,执行系统执行第一例外事件。处理前端还被用以设定从跨立于二快取线的快取线数据产生的每一指令的跨立位。在此实施例中,当被提出的指令中的跨立位被设定时,逾时侦测器会读取所有权快取存储器中对应的储存单元的下一个连续的储存单元,且当下一个连续的储存单元的逾时位被设定时,逾时侦测器会使被提出的指令被标记以产生第一例外事件。

[0009] 撷取系统会决定所有权索引为二进位计数值。当每一储存单元被输入进所有权队列时,二进位计数值会增加以作为所有权队列中的储存单元的总数量。此外,所有权索引的最高有效位会是缠绕位。处理器会还具有覆写侦测器。覆写侦测器用以依据被提出的指令中的所有权索引以读取所有权快取存储器中对应的储存单元中的缠绕位。且当对应储存单元的缠绕位并不匹配于被提出的指令的缠绕位时,覆写侦测器还用以使被提出的指令被标记以产生第一例外事件。当准备好退出的指令被标记以产生第一例外事件时,执行系统会执行第一例外事件。

[0010] 处理器会具有储存队列。储存队列会具有多个储存单元。每一储存单元被用以储存处理前端提出的储存指令,且每一储存单元用以储存执行系统决定的一个目标地址。执行系统会还具有储存管线。储存管线用以决定被调度以进行执行的每一储存指令的目标地址。且储存管线用以提供被决定的每一目标地址给储存队列的对应的储存单元与第二比较器。

[0011] 处理系统会依据被提出的指令中的所有索引存取所有权队列中对应的储存单元以设定对应的储存单元中的执行位。处理器会还具有逾时侦测器。逾时侦测器用以计算第二比较器决定的每匹配的一储存单元的执行位。且当匹配的储存单元中的任何的执行位被设定时,逾时侦测器会使对应于被决定的目标地址的储存指令被标记以产生第二例外事件。在此实施例中,当准备好退出的一储存指令被标记以产生第二例外事件时,执行系统会执行第二例外事件。在一实施例中,第二例外事件使执行系统令被标记以产生第二例外事件的储存指令退出,以刷新处理器,且以使撷取系统取得指令指标以从指令快取存储器取得储存指令后的指令。

[0012] 处理前端用以设定依据跨立于二快取线的快取线数据产生的每一指令中的跨立位。处理系统依据被提出的指令中的所有索引存取所有权队列中对应的储存单元以设定对应的储存单元中的执行位。此外,当被提出的指令中的跨立位被设定时,处理前端会用以设定对应的储存单元后的下一个连续的储存单元的执行位。处理器还具有逾时侦测器。逾时侦测器用以计算第二比较器决定的每匹配的一储存单元的执行位。且当匹配的储存单元中的任何的执行位被设定时,逾时侦测器会使对应于被决定的目标地址的储存指令被标记以产生待执行的第二例外事件。在此实施例中,当准备好退出的一储存指令被标记以产生第二例外事件时,执行系统会执行第二例外事件。第二例外事件使执行系统令被标记以产生第二例外事件的储存指令退出,以刷新处理器,且以使撷取系统取得指令指标以从指令快取存储器取得储存指令后的指令。

[0013] 依据一实施例,在依据快取线决定存储器所有权以侦测自修正程序码的方法中是先取得快取线。每一快取线具有快取线地址与快取线数据。决定每一取得的快取线的所有索引,且将快取线地址与所有权索引加入所有权队列中的一或多储存单元。当快取线地址被加入进储存单元时,于所述的方法中,还比较每一快取线地址与被提出的每一储存指令的每一目标地址,且标记任何的匹配的储存单元为逾时的。在所述的方法中,还具有,将快取线的快取线数据转译成指令,每一指令具有所有权索引,所有权索引是为了所有权队列中的储存单元被决定。储存单元储存有被转译成的指令所来自的快取线。所述的方法还具有提出指令以执行与决定被提出以执行的每一储存指令的目标地址。所述的方法还具有,当每一目标地址被决定时,比较每一目标地址与所有权队列中每一有效的储存单元的快取线地址,且标记任一匹配的一储存单元为逾时。所述的方法还具有,当准备好退出的指令具有匹配于所有权队列中的储存单元的所有索引,且所述的指令被标注为逾时的。

[0014] 所述的方法具有,刷新处理器,避免用以触发第一例外事件的指令退出,重新撷取用以产生第一例外事件的指令。

[0015] 所述的方法具有,当接收到新的快取线地址时,有效化所有权队列中的储存单元,标记所有权队列中的每一有效储存单元中的最后指令为最后指令,且当标记为最后指令的指令退出时,使所有权队列中的对应的储存单元无效化。

[0016] 所述的方法还具有,依据被提出的每一指令具有的所有索引以存取所有权队列中对应的储存单元,且当对应的储存单元被标记为逾时的时候,标记用以产生第一例外事件的指令,并为准备好退出且被标记以产生第一例外事件的每一指令执行第一例外事件。

[0017] 所述的方法具有,设定被转译自跨立于两快取线的快取线数据的每一指令的跨立位。所述的方法还具有,当每一指令被提出时,依据指令所具有的所有索引存取所有权队

列中对应的储存单元。且当跨立位被设定时,取得所有权队列中下一个连续的储存单元。当对应的储存单元被标记为逾时的时候,标记指令以产生第一例外事件。且当跨立位被设定且所有权队列中的下一个连续的储存单元被标记为逾时的时候,标记指令以产生第一例外事件。所述的方法还具有,执行准备好退出且被标记以产生第一例外事件的每一指令的第一例外事件。

[0018] 所述的方法还具有,重复地增加所有权索引如二进位计数值,二进位计数值具有一总计数量至少用以作为所有权队列中的储存单元的总数量。决定缠绕位,缠绕位为二进位计数值的最高有效位。每一被转译的指令的对应的缠绕位相同于所有权队列中储存有指令被转译所依据的快取线的储存单元所被决定的缠绕位。依据指令中的所有权索引取得所有权快取存储器中的储存单元,比较指令中的缠绕位与曲出的储存单元的缠绕位。当缠绕位并不匹配时,标记用以产生第一例外事件的指令,执行标记以产生第一例外事件的每一指令的第一例外事件。

[0019] 所述的方法还具有,当提出每一指令时,依据所有权索引取得所有权快取存储器的储存单元,且设定取出的储存单元的执行位。所述的方法还具有,在转译的过程中,设定依据跨立于二快取线的快取线数据产生的每一指令的跨立位。当提出每一指令时,当指令的跨立位被设定,取得所有权快取存储器中的下一个连续的储存单元,并设定下一个连续的储存单元的执行位。所述的方法具有,当比较被决定的目标地址于所有权队列中的每一有效的储存单元的快取线地址且匹配的储存单元被发现时,决定匹配的储存单元的执行位是否被设定。当匹配的储存单元的执行位被设定时,标记关联于被决定的目标地址的储存指令以产生第二例外事件。所述的方法还具有,使被标记以产生第二例外事件且预备退出的储存指令退出且完成,以刷新处理器,且通过取得储存指令于程序顺序之后的下一个指令,重新开始操作。

[0020] 本发明能够提高处理器的效能。

附图说明

[0021] 通过以下的叙述与图式,本发明的益处、特点与优点可以被更佳的理解。

[0022] 图1为依据一实施例实作以建立数据与指令间的所有权的一所有权队列结合一处理器的简化功能区块图。

[0023] 图2为依据一实施例实作的图1中的所有权队列具有相对于其他所有权处理模块的接口的一简化功能区块图。

[0024] 图3为依据一实施例中图1的处理前端的运算的流程示意图。

[0025] 图4为依据一实施例中所有权与例外事件处理的流程示意图。

[0026] 图5为依据一实施例中执行、退出与例外事件处理的流程示意图。

[0027] 其中,附图中符号的简单说明如下:

[0028] 100:处理器;101:所有权队列;102:系统存储器;103:预撷取模块;104:处理前端;105:指令快取存储器;106:执行系统;107:撷取模块;109:解码器;111:循环队列;113:循环侦测器;115:指令转译器;117:暂存器别名表;118:微操作;119:分支侦测器;121:重排序缓冲器;123:排程器;125:执行单元;127:储存队列;129:储存管线;130:数据快取存储器;131:其他元件;135:退出模块;137、139:逾时侦测比较器;141:覆写侦测器;143、145:逾时

侦测器;CA:快取线地址;DA:目标地址;EXB:执行位;L、T1、T2:栏位;IP:指令指标;OWNI:所有权索引;SDB:跨立位;STB:逾时位;UOP、UOPX:微操作;WB:缠绕位。

具体实施方式

[0029] 发明人发现了由于自修正程序码所产生的存储器所有权的问题。他们开发出了依据快取线建立存储器所有权的所有权队列以侦测自修正程序码。

[0030] 图1是处理器100结合一所有权队列(OWNQ)101的一个简化的功能区块图。所有权队列101依据一实施例而实作以建立数据与指令间的所有权。处理器100的标准指令集架构(instruction set architecture, ISA)可以是一个x86宏(macro)架构。此x86宏架构可以正确地执行被设计成执行于一个x86处理器的大部分的应用程序。一个应用程序的预期结果被取得时,应用程序算是被正确地执行。尤其是,处理器100执行x86指令集中的指令,且具有x86使用者可视的暂存器集。但是本发明并不限制于x86架构,处理器100可依据本领域普通技术人员理解的其他可替换的指令集架构。如图所示,处理器100耦接一外部系统存储器102。外部系统存储器102用以储存软件程序、应用程序、数据与本领域普通技术人员所理解的其他数据。处理器100可以具有一总线接口单元(bus interface unit, BIU)或相仿的元件(未绘示)以耦接系统存储器102。在一系统晶片的架构中,处理器100、系统存储器102与其它处理功能模块(未绘示)可以结合于一共用的集成电路。

[0031] 处理器100具有一处理系统。处理系统具有处理前端104与执行系统106以及其他于后续说明的处理模块。处理前端104具有一信息预撷取(PREFETCH)引擎103、一指令快取存储器(ICACHE)105、一撷取单元107、一解码器109、一循环队列(LQ)111、一指令转译器(XLATE)115、一暂存器别名表(RAT)117与一分支预测器119。执行系统106一般具有一重排序缓冲器(ROB)121、一排程器123(亦称保留站)、执行单元125与一储存队列127。执行单元125具有至少一储存管线129以及其他的执行单元131。执行单元131例如为一或多个整数(INT)单元、一或多个浮点数(或媒体)单元或至少一负载管线。在一实施例中,负载管线与储存管线可以被结合进一个存储器排序缓冲器(MOB)(未绘示)或相仿的元件。储存管线129可以还耦接一数据快取存储器(DCACHE)130。数据快取存储器130具有一或多个级别的数据快取存储器,例如一第一级别(L1)的快取存储器或一第二级别(L2)的快取存储器等等。数据快取存储器130可以也耦接系统存储器102。如图所示,重排序缓冲器121还具有一退出模块135,相关细节请容后再叙。

[0032] 其他的所有权逻辑与电路连同所有权队列101一同被提供,以进行所有权决定与侦测自修正程序码。以下进行相关细节的介绍。所述的其他的所有权逻辑与电路具有一第一逾时侦测比较器(STALE DETECT COMPARATOR1)137、一第二逾时侦测比较器(STALE DETECT COMPARATOR2)139、一覆写侦测器141、一第一逾时侦测器(STALE DETECTOR1)143与一第二逾时侦测器(STALE DETECTOR2)145。

[0033] 在一般性的操作中,预撷取引擎103从系统存储器102撷取程序资讯并储存所述的资讯进指令快取存储器105的快取线中。每一快取线可以具有一预设长度。所述的预设长度例如为64字节(byte)。快取线的尺寸可以是任意的且在其他的架构之下可以是不同的。撷取单元107从指令快取存储器105取得每一快取线且提供快取线数据给解码器109以将所述的数据解析成指令信息。解码器109将快取线数据分割且格式化成指令与对应于指令的信

息,例如运算域或相仿的信息。举例来说,在处理器100支持x86指令集架构的情况下,所述的指令例如为x86指令。此处提到的每一指令集架构例如是一宏指令或依据处理器100所支持的指令集的一宏操作。解码器109所提供的宏操作接着被加入循环队列111,且被提供至指令转译器115。指令转译器115将每一宏操作转译成一或多个对应的微指令或称微操作(micro operations,uop)。所述的微指令或微操作依据处理器100的原生指令集编排而成。当每一微操作被提供至重排序缓冲器121时,一指令指标(IP)也被决定且连同每一微操作被提供。微操作被提供至暂存器别名表117。暂存器别名表117用以依据每一微操作的程序排序、运算域源或重命名信息,产生每一微操作的相依信息。

[0034] 来自暂存器别名表117的每一微操作(连同相关联的信息)依据程序顺序被加入至重排序缓冲器121,且被加入至排程器123。排程器123具有至少一个队列,所述的队列用以储存接收自暂存器别名表117的每一微操作以及其相依信息。当微操作已准备好被执行时,排程器123调度接收到的微操作至对应的执行单元125。储存微操作被提供至储存管线129以进行处理,而所有其他的指令类型被提供至其他执行单元131中的适当的单元(例如整数指令被提供至整数执行单元,媒体指令被提供至媒体执行单元,等等)。当所有的相依关系被解决了,一微操作被认为是准备好进行执行。连同调度一微操作,暂存器别名表117将重排序缓冲器121的一储存单元安排给所述的微操作。因此,所述的微操作按程序顺序被分配至重排序缓冲器121中。重排序缓冲器121例如被编排成一循环队列,以确保所述的微操作依据程序顺序退出。暂存器别名表117也将对应的指令指标连同对应的相依信息提供给重排序缓冲器121,以将指令指标连同对应的相依信息储存于储存有微操作的储存运算元与结果的储存单元。在一实施例中,一个单独的物理暂存器堆(PRF)(未绘示)可以被包含进来。暂存器别名表117可以还将物理暂存器堆中的一或多个物理暂存器分配或映射给每一个微操作,以储存运算元与结果。

[0035] 执行单元211的结果例如被传回给重排序缓冲器121。重排序缓冲器121更新对应的栏位且/或更新架构暂存器(architectural register)或相仿的元件。在一物理暂存器堆的实施例中,重排序缓冲器121具有指标,指标用以更新物理暂存器堆中对应的暂存器。在一实施例中,暂存器别名表117将架构暂存器映射至物理暂存器堆中的物理暂存器,并更新重排序缓冲器121中对应于微操作的指标或其他相仿的信息(未绘示)。重排序缓冲器121中的指标例如在执行中或在执行之后被更新,且指标用以于运算中更新物理暂存器堆的暂存器中的内容。重排序缓冲器121中的退出模块135最后依据程序码顺序令微操作退出,以确保适当的运算与软件程序或应用程序原本的指令一致。当一微操作被标记或者是指示有一例外事件时,退出模块135依据例外事件的类型采取适当的行动。相关细节请见以下详述。

[0036] 被加入至储存管线129以进行运算的储存微操作也被加入至储存队列127中的对应的一个储存单元。当最初被从暂存器别名表117加入时,储存微操作的运算元的地址可能是不被确知的。储存微操作的运算元的地址包含目标地址(DA)。当储存管线129决定被执行的一储存微操作的目标地址,储存管线129提供目标地址至储存队列127中的对应的储存单元。

[0037] 分支预测器119侦测由解码器109提供的与/或循环队列111中的分支宏操作输出,且分支预测器119依据分支是否被采用而产生分支预测结果。分支预测器119与撷取单元

107进行通讯。撷取单元107依据分支预测结果可以在指令快取存储器105中分支至不同的位置。撷取单元107与预撷取引擎103也彼此进行通讯。因此,当分支位置并不位于指令快取存储器105中时,预撷取引擎103从系统存储器102取得对应的位置,以输入进指令快取存储器105。

[0038] 在正常的运算当中,来自解码器109的宏操作被经由循环队列111缓冲并被提供至指令转译器115。当循环侦测器113判断出循环中的指令多次递回,所述的循环例如全部位于循环队列111或者是至少部分位于循环队列111,循环侦测器113辨识一循环,所述循环中的指令重复地从循环队列111中被取出而不是从指令快取存储器105中被取出。在一实施例中,当一预设数量的循环递回发生时,循环侦测器113侦测到一循环。在一具体的实施例中,递回个数为24,但其他合适的递回个数也可以被使用。在一实施例中,循环侦测器113假设循环会无限的持续,因此循环侦测器113持续重复的循环运算直到预测不正确(循环分支不被采用),此时,系统被刷新,且撷取单元107开始在指令快取存储器105的循环之后的下一个位置(或可能是另一个分支位置)取得信息。

[0039] 在循环侦测器113已侦测到一循环的情况下,撷取单元107会持续地取得且将快取线加入至解码器109的缓冲器,直到缓冲器被填满,而撷取运算会暂时地停住。在一实施例中,当循环侦测器113侦测到一循环时,撷取单元107重复地取得循环中的快取线。于另一实施例中,撷取单元107会被通知循环侦测器113已侦测到一循环,且撷取单元107会开始读取循环外的数据。例如,撷取单元107会开始读取循环的下一个连续的位置。不论哪一个情况中,当一个循环进行中,解码器109会被填满。

[0040] 当撷取单元107将快取线数据加入解码器109时,撷取单元107也将对应的快取线地址(CA)加入所有权队列101中的一储存单元并标记此储存单元为有效的。所有权队列101会是被编排为循环缓冲器或相仿的结构,所有权队列101会具有加入指标与推出指标以区别已被分配的储存单元与已被解除分配的储存单元。于另一实施例中,所有权队列101中的每一储存单元具有一有效位或一有效值以区别有效储存单元与无效储存单元。其中,每一被加入所有权队列101的新的储存单元的有效位被设定。在一实施例中,撷取单元107决定一所有权索引(OWNI)与一缠绕(wrap)位(WB)。所有权索引与缠绕位对应于快取线的快取线地址,且对应的所有权索引值与缠绕位值被连同快取线地址一同加入所有权队列101中对应的储存单元。所有权索引唯一地定义出所有权队列101中的每一储存单元。在此缠绕位被用以侦测所有权队列101中的一覆写动作。

[0041] 暂存器别名表117用以依据对应的所有权索引识别每一快取线中的最后的微操作,并用以标注所述的快取线的微操作为最后一个的微操作,以使此信息被提供至重排序缓冲器121。当退出模块135退出一个微操作时,退出模块135决定被退出的微操作是否被标注为所有权队列101中被给定的一快取线的最后一个微操作。若是,退出模块135指示所有权队列101推出对应的储存单元或使所有权队列中的对应的储存单元无效。

[0042] 当撷取单元107将每一新的快取线地址加入进所有权队列101中的一储存单元时,快取线地址也被提供至第一逾时侦测比较器137的输入端。逾时侦测比较器137也自储存队列127中读取每一有效的目标地址(DA),并比较每一目标地址与新的快取线地址,以决定是否有所匹配者。逾时侦测比较器137可以被视为一种新储存单元的比较器。当快取线地址与任何的目标地址相匹配,所有权队列101中的储存单元的对应的一个逾时位被设定。逾时位

STB指示一储存微操作与快取线命中彼此,亦即,储存指令已经修改快取线或者是储存指令将会修改快取线。当一储存指令与储存在所有权队列101一有效储存单元的一快取线命中彼此或彼此冲突,任何依据此快取线产生的指令会是无效的。当逾时位STB被设定的时候,任何来自于所述的快取线的微操作可能是无效的(也就是逾时的)。

[0043] 所有权索引值与缠绕位值更被加进或关联至提供到解码器109的对应的快取线数据。解码器109具有每一宏操作的一对应的缠绕位值与所有权索引值以识别出是由解码器109从哪个宏操作对应的快取线取得。当多个宏操作被自同一条快取线取出时,同样的缠绕位与所有权索引被指派给来自同一条快取线的每一宏操作。在一实施例中,宏操作并未与数据快取存储器105中的快取线对齐时,每一宏操作还具有一跨立位SDB。跨立位SDB用以识别出一宏指令跨立于两条不同的快取线的情况。也就是说,一宏指令开始于其中一条快取线并结束于下一个连续的快取线。当这样的情况发生时,解码器109加入第一条线的所有权索引并设定宏操作的跨立位为真。当宏操作被包含进一单一的快取线,跨立位被设定为假。当被加入进指令转译器115时,每一宏操作具有对应的缠绕位、所有权索引与跨立位。当一个位或一个栏位被设定为真或假时,所述的位或栏位(具有至少一个位)被设定为逻辑“1”以设定为真,且被设定为逻辑“0”以设定为假。

[0044] 指令转译器115将每一宏操作转译成一或多个微操作。在转译的过程当中,由一宏操作产生的每一微操作同样具有与所来自的宏操作一样的缠绕位值、所有权索引值与跨立位值。因此,当一个宏操作被转译成三个个别的微操作时,三个微操作中的每一具有与原本的宏操作相同的缠绕位值、所有权索引值与跨立位值。当传输经过暂存器别名表117时,缠绕位值、所有权索引值与跨立位值还是被保留于每一微操作。

[0045] 一示范性微操作uopx示于图1中的118,且示范性微操作uopx用以被暂存器别名表117推出与被加入进重排序缓冲器121及排程器123中的为处理器100所定义的任意一个微操作。每一微操作具有多个栏位以便于对应的微操作的运算或执行被处理器100的执行系统106所执行。一或多个栏位(未绘示)用以识别特定的指令与指令类型及其相关联的运算元,例如常数运算元、地址、储存位置与暂存器索引等等的。其他的栏位被提供以储存指令指标IP、缠绕位WB、所有权索引OWNI与跨立位SDB。如以下所述地,每一微操作还具有栏位T1以指示第一类型的例外事件,每一微操作还具有栏位T2以指示第二类型的例外事件,每一微操作还具有栏位L以指示指令是否是被暂存器别名表117标记为快取线中的最后指令。

[0046] 当每一微操作自暂存器别名表117中推出且被加入重排序缓冲器121与排程器123中时,暂存器别名表117依据微操作的所有权索引OWNI的索引值存取所有权队列101中对应的储存单元,并设定取出的储存单元中的一个执行位EXB。当微操作的跨立位为真以指示其为跨立指令时,暂存器别名表117设定所有权队列101中下一个连续的储存单元的执行位。除此之外,储存单元的执行位用以侦测一未被侦测为逾时的储存微操作的之后的命中。

[0047] 当每一微操作被自暂存器别名表117中输出时,覆写侦测器141通过微操作所具有的所有权索引值存取所有权队列101中对应的储存单元,且覆写侦测器141通过微操作所具有的所有权索引值读取被取出的储存单元的缠绕位值。当微操作的缠绕位值并不匹配于所有权队列101中的对应的储存单元的缠绕位值时,一覆写作动曾经发生,且覆写侦测器141设置微操作的位T1(或是标记栏位T1为真)以指示一第一例外事件或是第一类型的例外事件于微操作被退出时被执行。在一实施例中,在微操作被加入重排序缓冲器121前,位T1于

微操作被提出之时可以被覆写侦测器141设定。在另一实施例中,当被加入至重排序缓冲器121时或当被加入至重排序缓冲器121之后,重排序缓冲器121中的储存单元的位T1被覆写侦测器141设定或是被重排序缓冲器121依据覆写侦测器141的指示设定。一个缠绕位不匹配通常指示一循环引发所有权队列101中的一覆写,以至于对应的快取线不再侦测自修正程序码。退出模块135侦测T1被设定以指示对应于重排序缓冲器121的微操作储存单元的微操作被标记成第一类型的例外事件。覆写意味着所有权队列101中的一个储存单元被覆写而使得关联于对应快取线的自修正程序码可能变得无法侦测而导致不正确的结果。更进一步地来说,第一类型的例外事件刷新了机器以预防不正确的状况。

[0048] 当每一微操作被自暂存器别名表117中输出时,第一逾时侦测器143依据微操作所具有的所有权索引存取所有权队列101中对应的储存单元,且第一逾时侦测器143依据微操作所具有的所有权索引读取取出的储存单元的逾时位。当微操作的跨立位为真时,第一逾时侦测器43还读取所有权队列101中的下一个连续的储存单元的逾时位STB。当所有权队列101中的此逾时位STB为真,或是微操作如跨立位SDB所示地为跨立指令且所有权队列101中的下一个连续的储存单元的逾时位为真,则第一逾时侦测器143通过设定栏位T1为真(或是通过设定位T1)以标记微操作成第一类型的例外事件。逾时侦测器143可以是指用以当指令被提出时,用以侦测可能的无效指令的提交逾时侦测器。如覆写侦测器141的例子,栏位T1于被加入进重排序缓冲器121中的一储存单元前、于被加入进重排序缓冲器121中的一储存单元时、或于被加入进重排序缓冲器121中的一储存单元后被设定为真,可以是由第一逾时侦测器143或是由重排序缓冲器121设定为真。如前述地,逾时位STB用以指示快取线已被一储存微操作修改,因此指令可能是无效的。

[0049] 每当储存管线129产生一目标地址(DA)给对应的一储存微操作时,目标地址除了被提供以更新储存队列127中对应的一储存单元之外,目标地址还提供给第二逾时侦测比较器139的一输入端。逾时侦测比较器139还存取所有有效的快取线地址且比较新的目标地址于所有权队列101中每一有效的快取线地址。逾时侦测比较器139可做为新目标地址的比较器。当有一相匹配结果时,逾时侦测比较器139设定所有权队列101中的对应的储存单元的逾时位为真。此外,当逾时侦测比较器139侦测到有一相匹配结果时,对应的所有权索引被提供至第二逾时侦测器145的一输入端。逾时侦测器145存取所有权队列101中对应的储存单元,且读取此储存单元中的执行位EXB。当储存单元的执行位EXB被设定为真,则逾时侦测器145使重排序缓冲器121中的储存微操作储存单元被标记成第二例外事件类型或者是第二类型的例外事件,这是通过设定储存单元的栏位T2为真。逾时侦测器145可做为用以侦测正执行中的可能的无效指令的执行中逾时侦测器。逾时侦测器145可以直接存取重排序缓冲器121中的储存微操作储存单元以设定T2,或者逾时侦测器145可以指示重排序缓冲器121以设定T2。

[0050] 关联于重排序缓冲器121中的一特定的微操作uopx的示范性的微操作118简化后被示于图1中。每一个别的微操作储存单元具有栏位T1以指示第一类型的例外事件,且每一个别的微操作储存单元具有栏位T2以指示第二类型的例外事件,每一个别的微操作储存单元具有栏位L以指示指令是否为被暂存器别名表117标记的快取线的最后一个微操作指令。当微操作为一快取线的最后一个微操作,栏位L被设定为真,反之则栏位L被设定为假。退出模块135侦测重排序缓冲器121的微操作的储存单元的栏位T1与栏位T2,且退出模块135执

行或初始化对应的例外常式(routine)或者是程序。然而,包含储存微操作的任何的微操作都可能被标记成第一类型的例外事件,但只有储存微操作会被标记成第二类型的例外事件。

[0051] 退出模块135侦测每一微操作于何时准备好退出,例如当微操作为重排序缓冲器121中最老的指令时。当一个微操作准备好退出时,退出模块135还侦测对应的微操作的储存单元中的栏位T1、栏位T2与栏位L。当一微操作的栏位T1为真时,退出模块135产生第一类型的例外事件给所述的微操作,且当栏位T2为真时,退出模块135产生第二类型的例外事件给所述的微操作。当栏位T1与栏位T2为假且栏位L为真时,退出模块135指示所有权队列101推出所有权队列101中对应的储存单元或是使所有权队列101中的所述的储存单元无效化,而有效率地从所有权队列101中移除完整的快取线。

[0052] 当重排序缓冲器121中最老的微操作(也就是将要退出)被指示为第一类型的例外事件,重排序缓冲器121在处理器100中广播一对应的例外事件讯号,且处理系统被刷新。在这样的情况下,执行管线中的任何的宏操作与微操作被有效率地无效化,包括导致所述的例外事件的微操作。当第一类型的例外事件发生时,所有未退出的微操作被刷新,包括储存队列127中任何未退出的储存微操作。退出的储存微操作仍持续停留在储存队列127中,直到其数据被提交给存储器架构(如数据快取存储器130与/或系统存储器102)。导致第一类型的例外事件的微操作并不被允许退出,微操作记录于暂存器重排缓冲器121中对应的指令指标可被使用以存取微操作于指令快取存储器105中的地址。预撷取引擎103与撷取单元107暂时的停住。处理器100中断于处理器100的一微程序码唯读存储器(未绘示)中的例外常式,且对应的例外程序码用以指示例外事件的类型。一旦处理系统被刷新时,例外常式取得指令指标且将指令指标传给撷取单元107以重新撷取关联于导致例外事件的微操作的宏操作。

[0053] 储存微操作的第二类型的例外事件相仿于其他类型的微操作的第一类型的例外事件。在此情况中,储存微操作被允许退出,使储存微操作完成了其运算且更新其目标地址所指出的存储器位置。因为存储器位置先被指令快取存储器105所拥有,且储存微操作是需要被处理器100中的数据快取存储器130所拥有的数据运算,因此一监控(snoop)单元被初始化以使指令快取存储器105中对应的快取线无效化。存储器修正与无效化被确保会在例外事件时发生。相仿于第一类型的例外事件,第二类型的例外事件的例外常式刷新机器,且存取与传递指令指标至撷取单元,以于所述的位置重新开始。因为导致第二类型的例外事件的储存微操作被允许完成,指令指标被增加至储存微操作于指令快取存储器105之后的下一指令,且运算由储存指令之后的所述的位置继续。

[0054] 图2为依据一实施例实作的图1中的所有权队列101具有对应于其他所有权处理模块的接口的一简化功能区块图。所有权队列101具有多个储存单元。每一储存单元具有一栏位WRAP以储存缠绕位。每一储存单元具有一栏位OWNI以储存对应的一索引值。每一储存单元具有一执行栏位以储存对应的一执行位。每一储存单元具有一有效栏位以储存有效位。每一储存单元具有一快取线地址栏位以储存对应的一快取线地址。每一储存单元具有一逾时栏位以储存对应的逾时位。

[0055] 在一实施例中,所有权索引为一计数值。当每一储存单元被加入进所有权队列101时,所述的计数值增加。为了确保所有权队列101中的每一储存单元的所有权索引具有一独

特的索引值,所有权索引的位数B对应于所有权队列101中的储存单元的个数N,例如 $2^B \geq N$ 。在一个例子中,如图2所示,所有权队列101中的储存单元的数量为 $N=32$,且所有权索引的位为5。在一实施例中,撷取单元107以类似方式决定缠绕位,缠绕位做为所有权索引的一个额外的最高有效位。在这样的情况下,当所有权索引从0计数到一最大值时,缠绕位是0b(b用以代表一二进位数),其中最大值用以指示所有权队列101中的储存单元的总个数。当所有权被重设为0且重新累加至最大值时,且缠绕位是1b。换句话说,对于所有权队列101每一次完整的传递(pass),缠绕位WB被切换于两值之间。对于所有权索引的位数B,储存单元的总数可以少于储存单元最大可能的个数。举例来说,对总数量为26的储存单元来说,在第一次递回时(OWNI从十进位的0累计至十进位的25,且WB为0),WB|OWNI从0|00000b计数至0|11001b。接着,在第二次递回中(OWNI从十进位的0计数至十进位的25且WB为1),从1|00000b计数至1|11001b。后续依上述的方式重复。

[0056] 如前述地,撷取单元107将一新的快取线地址CA插入快取线地址栏位,并设置有效栏位中对应的有效位,并决定与插入对应的所有权索引至栏位OWNI,并决定与插入对应的缠绕位WB至栏位WRAP。新被加入的快取线地址被提供至一逾时侦测比较器137的一输入端。逾时侦测比较器137也从储存队列127接收目标地址DA。当新的快取线地址与来自储存队列的任何目标地址之间有相匹配的结果时,逾时栏位中对应的逾时位被设定为真。当每一微操作被自暂存器别名表117提出时,它依据微操作的所有权索引存取所有权队列101中对应的储存单元以设定储存单元对应的执行位EXB。此外,当微操作的跨立位被设定以指示一跨立微操作时,暂存器别名表117存取所有权队列101中的下一个储存单元并设定该储存单元对应的执行位。当一快取线的最后的微操作退出时,暂存器重排缓冲器121存取所有权队列101中对应的储存单元并重置或清除有效位。

[0057] 储存管线129决定每一储存微操作的目标地址并储存目标地址于储存队列127的对应的储存单元。目标地址也被提供至第二逾时侦测比较器139。第二逾时侦测比较器139从所有权队列101中存取快取线地址CA。当新被决定的目标地址匹配来自所有权队列101的快取线地址中的任何一个时,逾时侦测比较器139设定所有权队列101中的对应的储存单元的逾时位。此外,匹配的所有权索引的索引值被提供至第二逾时侦测器145。第二逾时侦测器145依据所有权索引值存取所有权队列101中对应的储存单元以存取对应的执行位EXB。当所有权队列101中对应的储存单元的执行位被设定为真,逾时侦测器145将重排序缓冲器121中冲突的储存微操作的储存单元标记(或是令其被标记)成第二类型的例外事件。

[0058] 如前述地,覆写侦测器141从暂存器别名表117提出的微操作接收缠绕位值与所有权索引值,且覆写侦测器141依据所有权索引从所有权队列101中的对应的储存单元存取缠绕位值。当所有权队列101中对应的储存单元的缠绕位WB不匹配于微操作的缠绕位,覆写侦测器141将微操作标记(或是令其被标记)成第一类型的例外事件。此外,第一逾时侦测器143从暂存器别名表117提出的微操作接收所有权索引值与跨立位值,且第一逾时侦测器143依据所有权索引值以存取所有权队列101中对应的储存单元的逾时位。当被暂存器别名表117提出的微操作中的跨立位为真以指示一跨立微操作,则逾时侦测器143存取所有权队列101中下一个连续的储存单元的逾时位。当被存取的储存单元的任何的逾时位被设定为真,逾时侦测器143将微操作标记(或是令其被标记)成第一类型的例外事件。

[0059] 一第一储存单元显示于所有权队列101的顶部,第一储存单元具有一缠绕位WB、一

所有权索引、一执行位EXB、一有效位、一对应的快取线地址CA_33与一逾时位STB。其中,缠绕位WB=1b。所有权索引的索引值为00000b。执行位的值为0b。有效位的值为1b。逾时位的值为0b。一所有权队列101中的第二储存单元位于第一储存单元的下方。第二储存单元具有一缠绕位WB、一所有权索引、一执行位EXB、一有效位、一对应的快取线地址CA_34与一逾时位STB。其中,缠绕位WB=1b。所有权索引的索引值为00001b。执行位的值为0b。有效位的值为1b。逾时位的值为0b。一所有权队列101中的第三储存单元位于第二储存单元的下方。第三储存单元具有一缠绕位WB、一所有权索引、一执行位EXB、一有效位、一对应的快取线地址CA_03与一逾时位STB。其中,缠绕位WB=0b。所有权索引的索引值为00010b。执行位的值为0b。有效位的值为0b。逾时位的值为0b。往所有权队列101的末端,最后五个储存单元分别具有快取线地址CA_28至CA_32与对应的所有权索引,分别具有所有权索引值11011b-11111b。具有快取线地址CA_28的储存单元还具有执行位、有效位与逾时位。执行位、有效位与逾时位的值皆为0b。而接下来分别具有快取线地址CA_29-CA_31的三个储存单元分别具有值为1b的储存位与值为1b的有效位。具有快取线地址CA_29的储存单元与具有快取线地址CA_31还具有值为0b的逾时位。而具有快取线地址CA_30的储存单元则具有值为1b的逾时位。具有快取线地址CA_32的最后储存单元为有效但尚未执行,且被标记为逾时的。

[0060] 在第一次传递中,快取线地址CA_1至快取线地址CA_32以具有值0b的缠绕位填满所有权队列101。在第二次传递刚开始时,第一次传递中最前面两个储存单元会被具有快取线地址CA_33与具有快取线地址CA_34与所有权索引值00000b所有权索引值00001b的储存单元给分别覆写,储存单元各具有值为1b的缠绕位WB。这些新的储存单元(33与34)为有效的,但尚未有任何微操作被执行。第三至第二十八个储存单元已被无效化(可能是完成再无效)。第二十九与第三十一储存单元为有效的,且每一具有至少一微操作正在执行中。第三十储存单元为有效的且具有至少一仍在执行中的微操作,但已被标记为逾时的。第三十二个储存单元并不具有从暂存器别名表117中被提出的微操作,因此第三十二个储存单元尚未被设定为执行,但是其逾时位已被设定以指示与一储存指令冲突或命中。

[0061] 当撷取单元107计数所有权索引值至11111b,同时缠绕位WB的值如所有权队列101中最后的具有快取线地址CA_32(例如第一次传递)的储存单元所指示的为0b,它设定缠绕位的值如具有快取线地址CA_33(开始第二次传递)的储存单元所指示的为1b且将所有权索引值重置为00000b重新开始计数。后续被撷取单元107读取的三十一个储存单元的缠绕位WB的值持续维持为1b直到所有权索引被重置为00000b,操作以如上述的方式重复。当一个循环被侦测到时,循环队列111不再自解码器109加入宏操作,撷取单元107仍然持续地自快取存储器105读取快取线至所有权队列101与解码器109,所有权队列101中的对应于循环指令的一储存单元可能被撷取单元107覆写。在这样的情况下,处理器100可能不能够再为所述的快取线侦测自修正程序码。被暂存器别名表117提出的而位于一循环中的微操作的缠绕位WB的值不再与所有权队列101中被覆写的储存单元的缠绕位的值一样。在被提出的微操作的缠绕位值与所有权队列101中对应的储存单元的缠绕位值不匹配的情况下,覆写侦测器141侦测被覆写的快取线并将微操作标记(或是令其被标记)成第一类型的例外事件。即使所有权队列101中的储存单元被标记为无效或被自队列中推出,这依然为真。一个无效的或被推出的储存单元持续停留在所有权队列101中直到被覆写。

[0062] 图3为依据一实施例中处理前端104的运算的流程示意图。在第一个区块301,快取

线(例如从系统存储器102)被读取且储存于指令快取存储器105中,例如是通过信息预撷取引擎103。在下一个区块303,下一条快取线的一个缠绕位与所有权索引值被决定,例如是通过撷取单元107,且这些信息连同快取线地址被加入所有权队列101中下一个可用的储存单元。撷取单元107还设定所有权队列101中的储存单元中的有效位。如前述地,所有权队列101例如被实施成一循环的缓冲器,所述的有效位用以决定任意时点中所有权队列101中的当前的有效储存单元。在一可替代的实施例中,加入指标与推出指标可被使用。

[0063] 如下一个区块305所示,当一个新的快取线地址被加入所有权队列101,新的快取线地址被比较于储存队列127中的每一个有效的目标地址。如下一个询问区块307所示,当有一命中被决定时,在区块309中,接收新的快取线地址的储存单元的逾时位STB被设定。在逾时位值被设定或者并没有命中时,所有权队列101的操作结束。

[0064] 如区块311所示,同时,当一个新的快取线地址被加入所有权队列101中时,对应的快取线数据连同缠绕位与所有权索引被加入解码器109。而在下一个区块313,解码器109解析快取线中的宏操作,且将宏运算所在的快取线的对应的缠绕位与所有权索引加入进每一宏运算。此外,解码器109决定宏操作是否跨立两条快取线,也就是说宏操作起始于一条快取线并结束于下一条连续的快取线。若是,则宏操作的跨立位被设定。此时,每一宏操作具有缠绕位值、所有权索引值与跨立位值。

[0065] 如区块315所示,宏操作接着被加入循环队列111,且如区块317所示,接着被加入指令转译器115。宏操作被转译成对应的微操作。如前述地,每一宏操作被转换成一或多个微操作。每一微操作具有被转译的宏操作的缠绕位值、所有权索引值与跨立位值。此时,每一微操作的指令指标也被指定为加入于微操作。在另一例子中,是在区块319或321时,指令指标被加入于每一微操作。在这些架构中的其中任一,指令指标最终连同每一微操作被加入重排序缓冲器121。在下一个区块319中,微操作被加入暂存器别名表117中,暂存器别名表117用以依据程序顺序、运算元与重命名信息产生每一微操作的相依信息。在区块321中,暂存器别名表117识别出且标注出位于一快取线最后的每一微操作,如前述的一实施例,是通过设定栏位L为真。此信息被传给重排序缓冲器121且被提供给重排序缓冲器121的对应的储存单元,因此退出模块135可以辨识每一快取线的指令何时被处理。接着,微操作被从暂存器别名表117中提出以进行执行以及后述的所有权与例外事件处理。

[0066] 图4为依据一实施例中所有权与例外事件处理的流程示意图。在第一个区块401,暂存器别名表117提出每一个微操作至重排序缓冲器121与排程器123。此外每一储存微操作还被加入储存队列127中。相关操作继续进行至区块402,自暂存器别名表117中提出的微操作的所有权索引用于存取所有权队列101中对应的储存单元。此操作已说明于前述的处理器100的多个功能方块的叙述中,但可以集中于共同的逻辑。当微操作被自暂存器别名表117中提出时,相关操作接着进行至三个不同的区块,区块403、区块405与区块411。

[0067] 在区块403中,储存单元的执行位EXB被设定。此外,若微操作的跨立位也为真,所有权队列101中的下一个连续的储存单元也被取出,且储存单元的执行位也被设定。此时,所有权队列101中被取出微操作的至少一快取线被标记为执行中,亦即,至少一快取线的微操作被提出以进行执行。在一或二执行位被设定后,流程图中的此分支被完成。

[0068] 在区块405,储存单元中对应的缠绕位WB被取得且被比较于微操作的缠绕位WB。当微操作的缠绕位WB与所有权队列101中的储存单元对应的缠绕位WB不匹配,如在下个询问

区块407,操作进行至区块409,而微操作标记成第一类型的例外事件(例如通过设定T1为真)。在标记(不匹配)之后或在缠绕位WB被判断为匹配,流程图的此分支的相关操作结束。

[0069] 在区块411中,所有权队列101中取出的储存单元的逾时位STB被取得。此外,当微操作的跨立位值为真,所有权队列101的下一个连续的储存单元的逾时位也被取得。在区块413中,判断逾时位是否被设置。当两个逾时位其中之一被设定时,相关操作进行至区块409,微操作标记成第一类型的例外事件(例如通过设定T1为真)。在于区块409中标记微操作作为第一类型的例外事件之后或当两个逾时位都未被设定时,流程图的此分支的操作结束。

[0070] 当如前述地准备好被执行时,被提出至排程器123的每一微操作最终会被调度至多个执行单元125中的对应的其中之一。还包含了如区块415所示地调度储存微操作至储存管线129。在下一个区块417,储存管线129决定储存微操作的目标地址且更新储存队列127中对应的储存单元。在下一个区块419,当每一新的目标地址被决定,目标地址被比较于所有权队列101中的有效快取线地址。在区块421中,判断新的目标地址是否匹配于有效的快取线地址。当新的目标地址并不匹配于所有权队列101中的有效的快取线地址任何一个,相关操作完成。

[0071] 当一个新的目标地址匹配于一个有效的快取线地址,相关操作进行至区块423,每个匹配的一储存单元的逾时位被设定。此外,匹配的储存单元的所有权索引被传至逾时侦测器145。在下一个区块425,逾时侦测器145依据被提供的所有权索引存取对应的储存单元以取得储存单元的执行位EXB。在下一个询问区块427中,当执行位EXB被决定为真,相关操作进行至区块429,冲突的储存微操作标记成第二类型的例外事件(例如通过设定T2为真)。在区块427中,当执行位EXB被决定为假或是在区块429标记储存微操作之后,操作结束。

[0072] 图5为依据一实施例中执行、退出与例外事件处理的流程示意图。在第一个区块501中,微操作从排程器123如前述地被调度至执行单元125。被调度的为操作会执行,但特定操作情况下并非如此。在下一个区块503,重排序缓冲器121的退出模块135识别出下一个要退出的微操作。在下一个询问区块505,待退出的微操作的栏位T1被决定是否被设定为真(例如由退出模块135决定)。若是,则相关操作进行至区块507,第一类型的例外事件被执行,包括刷新处理器100。此外,导致第一类型的例外事件的微操作被如前述地从指令快取存储器105中重新撷取。例外事件的处理操作完成。

[0073] 如下一个询问区块509,当T1不为真但T2被决定为真(例如经由退出模块135),相关操作进行至区块511,此时第二类型的例外事件被执行,储存微操作被允许完成与退出,且处理器100被刷新。在储存微操作开始例外事件之后,操作被重新开始于指令快取存储器105中的下一个指令。例外事件处理的相关操作于此被完成。在区块513,当T1与T2都不为真,微操作被允许退出。在区块514,当微操作的栏位L被设定为真,以指示为操作为快取线的最后一个微操作,则在区块515退出模块135指示所有权队列101以使对应的储存单元无效化,且操作被完成。所述的无效化例如是通过标记储存单元为无效,或者是推出所有权队列101中的储存单元堆迭中的储存单元。当栏位L为假,在指令被退出后,操作完成。

[0074] 前述的相关内容被以本领域普通技术人员能制造或使用本发明,如所提供关联于特定应用与必要条件的内容。尽管本发明已参照于某些相关的版本而被相当仔细的方式描述,其他的版本与变异为可行的且是经过仔细考虑过的。前述提及的实施例的多个变化形

对本领域普通技术人员来说会是显而易见的,且上述定义的一般性元则也可以适用于其他的实施例。例如此处所描述的电路可以被实作于适当的方式,例如逻辑装置或类似的电路。

[0075] 以上所述仅为本发明较佳实施例,然其并非用以限定本发明的范围,任何熟悉本项技术的人员,在不脱离本发明的精神和范围内,可在此基础上做进一步的改进和变化,因此本发明的保护范围当以本申请的权利要求书所界定的范围为准。

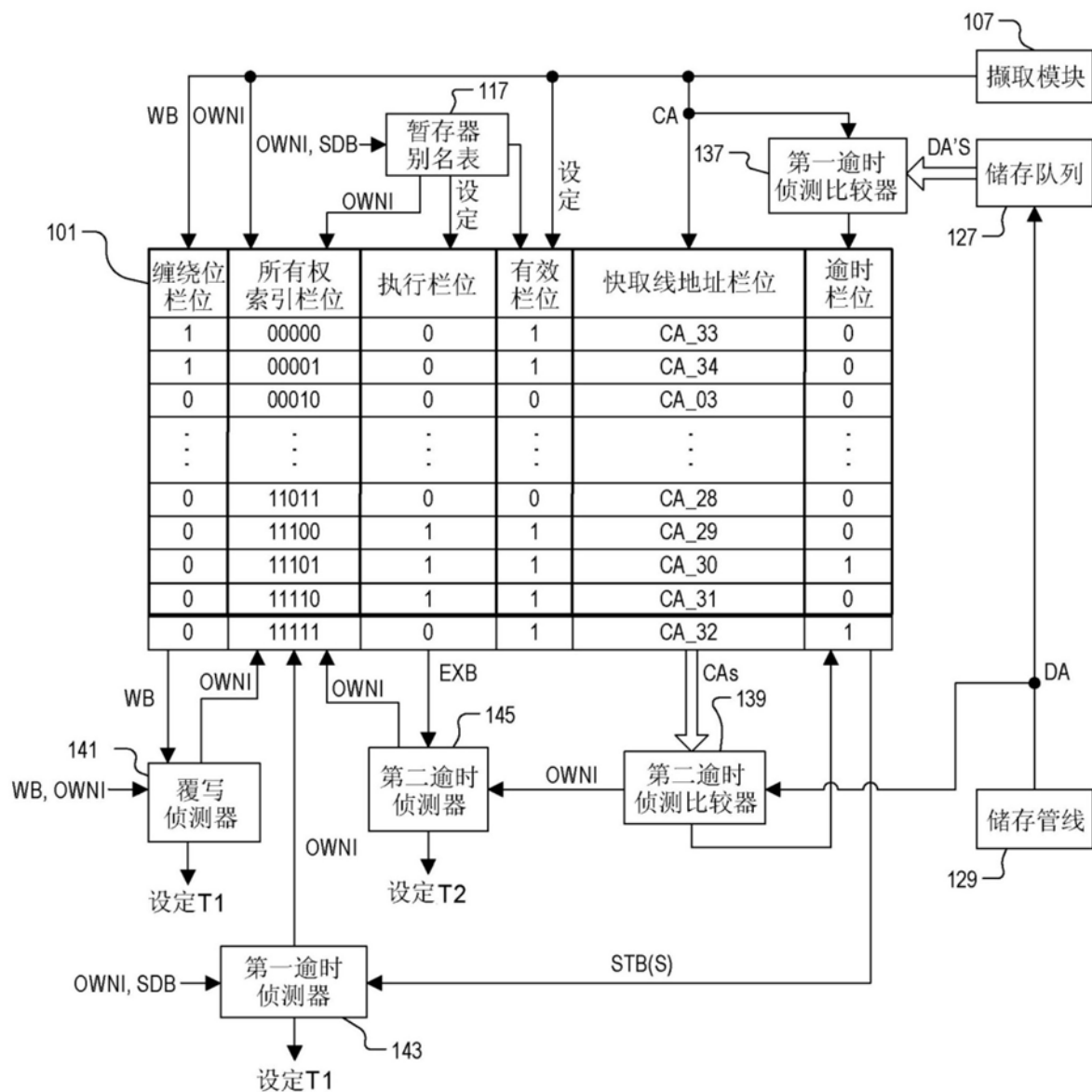


图2

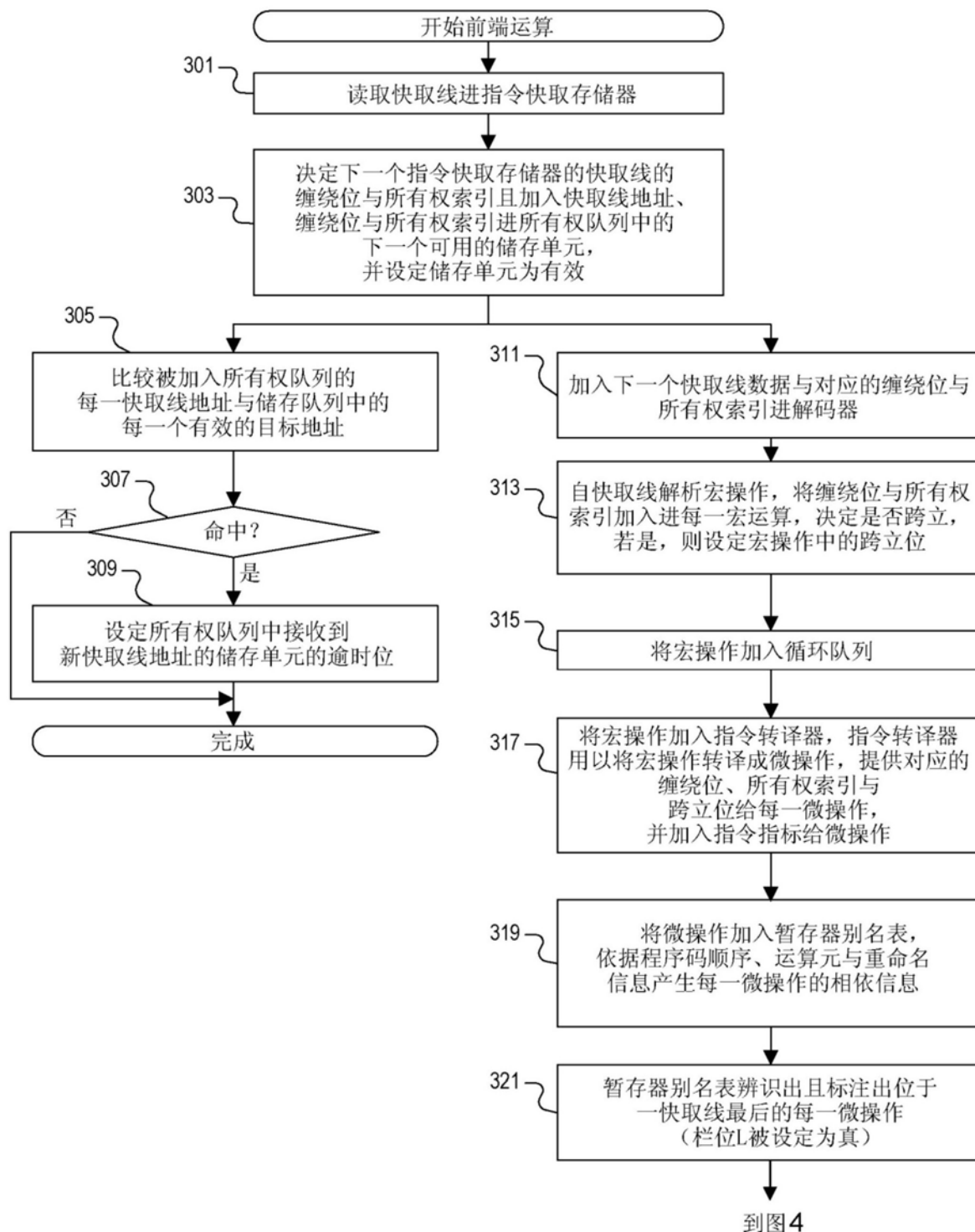


图3

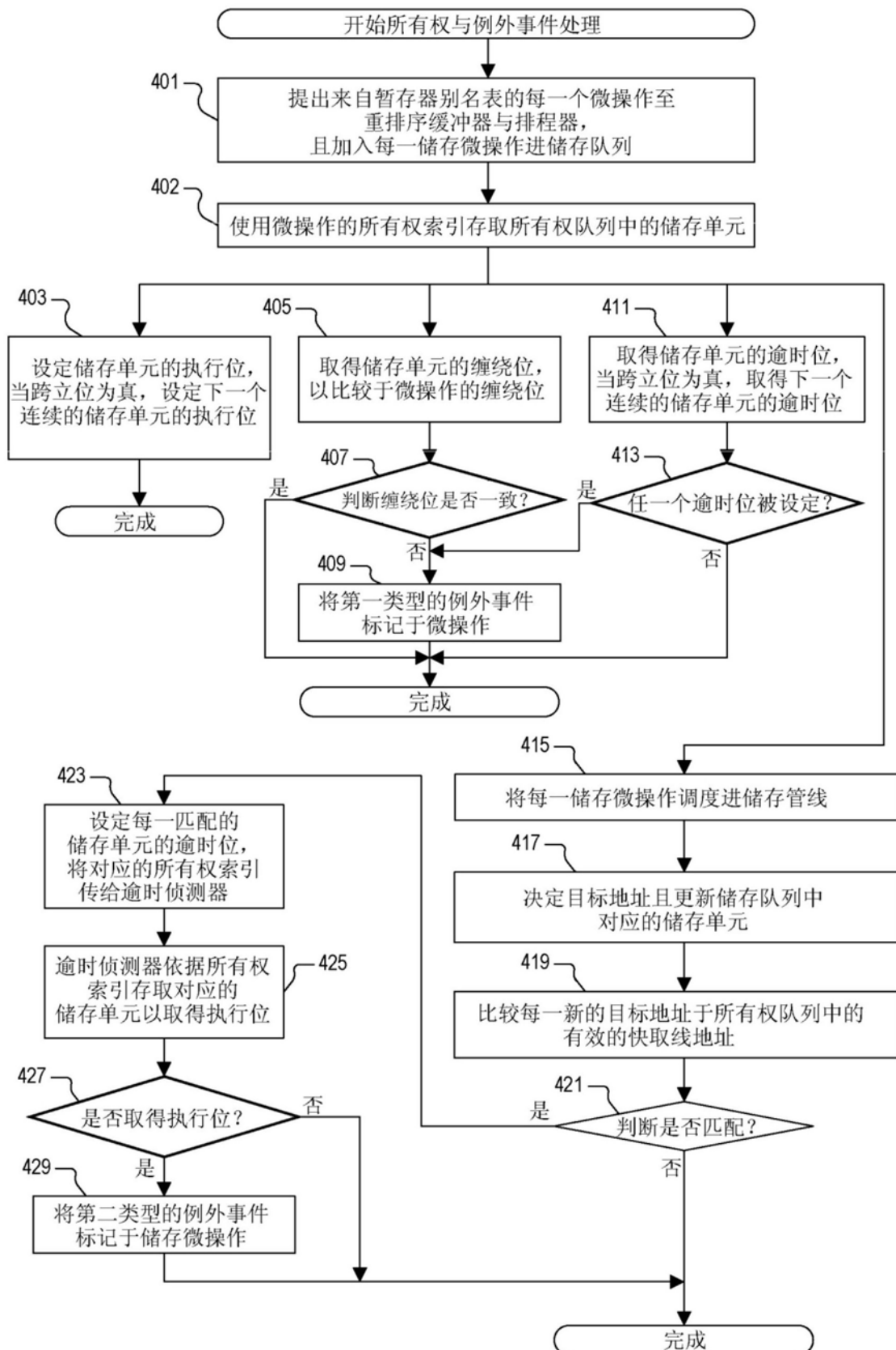


图4

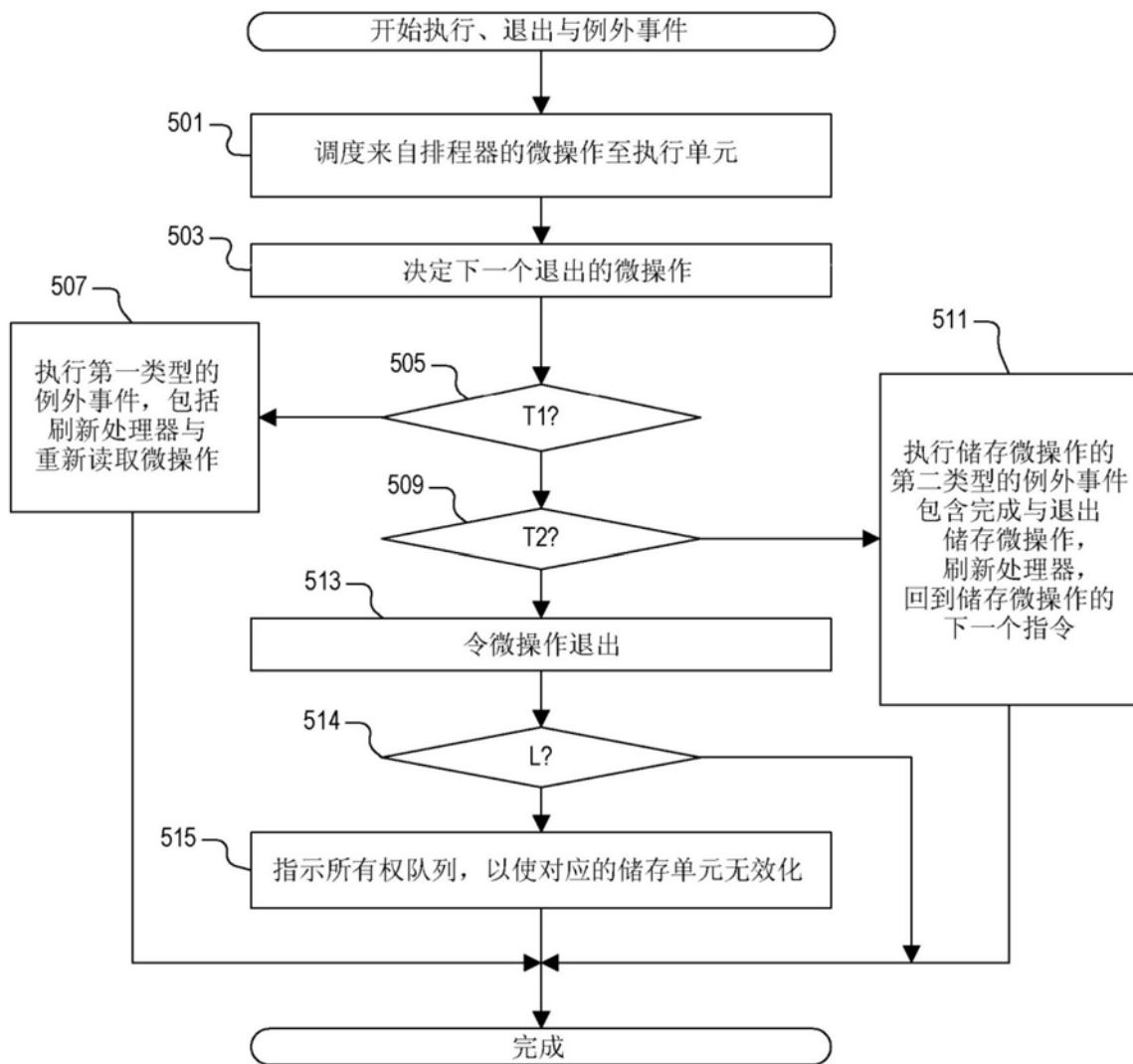


图5