(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105511839 A
(43)申请公布日 2016.04.20

(21)申请号 201510866716.4

(22)申请日 2015.12.01

(30)优先权数据
PCT/IB2014/003218 2014.12.14 IB

(71)申请人 上海兆芯集成电路有限公司
地址 201203 上海市浦东新区张江高科技园区金科路2537号301室

(72)发明人 吉拉德·M·卡尔 柯林·艾迪 葛兰·G·亨利

(74)专利代理机构 北京汇泽知识产权代理有限公司 11228
代理人 张瑾

(51)Int.Cl.
G06F 9/38(2006.01)

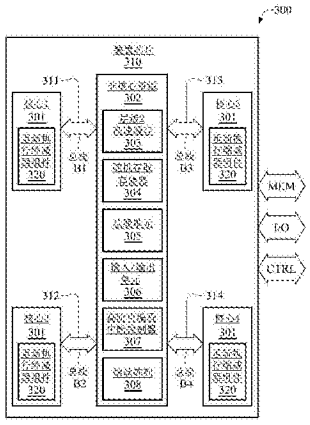
权利要求书3页 说明书15页 附图6页

(54)发明名称

用以改善在处理器中重新执行加载的装置与方法

(57)摘要

本发明提供一种用以改善在处理器中重新执行加载的装置与方法，所述装置包括第一保留站和第二保留站。第一保留站派送第一加载微指令，且若第一加载微指令是多个规定的加载微指令的其中一个，在总线进行侦测和指示。第二保留站在第一加载微指令派送后的第一数量的时钟周期后，派送和第一加载微指令相依的新微指令以进行执行，且若在总线上指示了，第一加载微指令是多个规定的加载微指令的其中一个，第二保留站缓存一或多个新微指令的派送，直到第一加载微指令取得操作数。非核心资源包括熔丝阵列以储存对应乱序处理器的规定的加载微指令，且在初始化时，乱序处理器存取熔丝阵列以决定规定的加载微指令。



CN 105511839 A

1.一种用以改善在一乱序处理器重新执行加载的装置，其特征在于，所述装置包括：

第一保留站，用以派送第一加载微指令，以及若所述第一加载微指令是指示从多个非核心资源的其中一个的多个规定的加载微指令的其中一个，用以在保留总线进行侦测和指示；

第二保留站，耦接至所述保留总线，且在所述第一加载微指令派送后的第一数量的时钟周期之后，用以派送和所述第一加载微指令相依的一或多个新的微指令以进行执行，以及若在所述保留总线上指示了，所述第一加载微指令是所述多个规定的加载微指令的其中一个，所述第二保留站用以缓存所述一或多个新的微指令的派送，直到所述第一加载微指令取得所述操作数；以及

所述多个非核心资源，包括：

熔丝阵列，用以储存对应所述乱序处理器的所述多个规定的加载微指令，其中在初始化时，所述乱序处理器存取所述熔丝阵列，以决定所述多个规定的加载微指令。

2.根据权利要求1所述的装置，其特征在于，所述乱序处理器包括多核心处理器，以及其中在所述多核心处理器的每一核心包括所述第一保留站和所述第二保留站。

3.根据权利要求2所述的装置，其特征在于，所述多个非核心资源包括所述熔丝阵列，以及其中所述熔丝阵列和所述每一核心一样被安置在相同的芯片上，但配置在所述每一核心之外。

4.根据权利要求2所述的装置，其特征在于，所述多个非核心资源未和所述多核心处理器一样被安置在相同的芯片上，以及其中所述多个非核心资源经由和所述每一核心一样被安置在相同的芯片上的总线被存取，但配置在所述每一核心之外。

5.根据权利要求1所述的装置，其特征在于，还包括：

执行单元，耦接至所述第一保留站，用以接收和执行所述第一加载微指令，以及若无接收到微指令以进行执行时，用以进入节能状态。

6.根据权利要求5所述的装置，其特征在于，若所述第一加载微指令非所述规定的加载微指令，当超过成功执行所需的所述第一数量的时钟周期，所述执行单元在对应非命中的总线上指示，所述第一加载微指令未成功执行，且启动所述一或多个新的微指令的重新执行。

7.根据权利要求6所述的装置，其特征在于，若所述第一加载微指令是所述规定的加载微指令，当超过成功执行所需的所述第一数量的时钟周期，所述执行单元不会指示所述第一加载微指令未成功执行，且预防所述一或多个新的微指令的重新执行。

8.一种用以改善重新执行加载的装置，其特征在于，所述装置包括：

多核心处理器，包括多个核心，其中每一所述多个核心包括：

第一保留站，用以派送第一加载微指令，以及若所述第一加载微指令是指示多个非核心资源的其中一个的多个规定的加载微指令的其中一个，用以在保留总线进行侦测和指示；

第二保留站，耦接至所述保留总线，且在所述第一加载微指令派送后的第一数量的时钟周期之后，用以派送和所述第一加载微指令相依的一或多个新的微指令以进行执行，以及若在所述保留总线上指示了，所述第一加载微指令是所述多个规定的加载微指令的其中一个，所述第二保留站用以缓存所述一或多个新的微指令的派送，直到所述第一加载微指

令取得所述操作数;以及

所述多个非核心资源,包括:

熔丝阵列,用以储存对应所述乱序处理器的所述多个规定的加载微指令,其中在初始化时,所述乱序处理器存取所述熔丝阵列,以决定所述多个规定的加载微指令。

9.根据权利要求8所述的装置,其特征在于,所述多核心处理器包括x86-兼容性多核心处理器。

10.根据权利要求8所述的装置,其特征在于,所述多个非核心资源包括所述熔丝阵列,以及其中所述熔丝阵列和所述每一核心一样被安置在相同的芯片上,但配置在所述每一核心之外。

11.根据权利要求8所述的装置,其特征在于,所述多个非核心资源未和所述多核心处理器一样被安置在相同的芯片上,以及其中所述多个非核心资源经由和所述每一核心一样被安置在相同的芯片上的总线被存取,但配置在所述每一核心之外。

12.根据权利要求8所述的装置,其特征在于,每一所述多个核心还包括:

执行单元,耦接至所述第一保留站,用以接收和执行所述第一加载微指令,以及若无接收到微指令以进行执行时,用以进入节能状态。

13.根据权利要求12所述的装置,其特征在于,若所述第一加载微指令非所述规定的加载微指令,当超过成功执行所需的所述第一数量的时钟周期,所述执行单元在对应非命中的总线上指示,所述第一加载微指令未成功执行,且启动所述一或多个新的微指令的重新执行。

14.根据权利要求13所述的装置,其特征在于,若所述第一加载微指令是所述规定的加载微指令,当超过成功执行所需的所述第一数量的时钟周期,所述执行单元不会指示所述第一加载微指令未成功执行,且预防所述一或多个新的微指令的重新执行。

15.一种用以改善在一乱序处理器重新执行加载的方法,其特征在于,所述方法包括:

配置多个非核心资源,其中所述多个非核心资源包括熔丝阵列,用以储存对应所述乱序处理器的所述多个规定的加载微指令;

经由第一保留站派送第一加载微指令,以及若所述第一加载微指令是指示多个非核心资源的其中一个的多个规定的加载微指令的其中一个,用以在保留总线进行侦测和指示;以及

经由耦接至所述保留总线的第二保留站,在所述第一加载微指令派送后的第一数量的时钟周期之后,派送和所述第一加载微指令相依的一或多个新的微指令以进行执行,以及若在所述保留总线上指示了,所述第一加载微指令是所述多个规定的加载微指令的其中一个,所述第二保留站用以缓存所述一或多个新的微指令的派送,直到所述第一加载微指令取得所述操作数。

16.根据权利要求15所述的方法,其特征在于,所述乱序处理器包括多核心处理器,以及其中在所述多核心处理器的每一核心包括所述第一保留站和所述第二保留站。

17.根据权利要求16所述的方法,其特征在于,所述多个非核心资源包括所述熔丝阵列,以及其中所述熔丝阵列和所述每一核心一样被安置在相同的芯片上,但配置在所述每一核心之外。

18.根据权利要求16所述的方法,其特征在于,所述多个非核心资源未和所述多核心处

理器一样被安置在相同的芯片上，以及其中所述多个非核心资源经由和所述每一核心一样被安置在相同的芯片上的总线被存取，但配置在所述每一核心之外。

19.根据权利要求15所述的方法，其特征在于，还包括：

经由耦接至所述第一保留站的执行单元，接收和执行所述第一加载微指令，以及若无接收到微指令以进行执行时，进入节能状态。

20.根据权利要求19所述的方法，其特征在于，若所述第一加载微指令非所述规定的加载微指令，当超过成功执行所需的所述第一数量的时钟周期，在对应非命中的总线上指示，所述第一加载微指令未成功执行，且启动所述一或多个新的微指令的重新执行。

21.根据权利要求20所述的方法，其特征在于，若所述第一加载微指令是所述规定的加载微指令，当超过成功执行所需的所述第一数量的时钟周期，不会指示所述第一加载微指令未成功执行，且预防所述一或多个新的微指令的重新执行。

用以改善在处理器中重新执行加载的装置与方法

技术领域

[0001] 本发明主要有关于一微电子领域的技术，特别有关改善在一乱序(out-of-order)处理器中重新执行加载的一节能机制。

背景技术

[0002] 机体装置技术在过去四十年迅速地发展。尤其在微处理器的发展，从4位、单一指令、10微米装置开始，随者半导体生产技术的进步，使得设计者能够设计出在架构和密度越来越复杂的装置。在80和90年代所谓的管线微处理器(pipeline microprocessor)和超纯量处理器(superscalar)，发展成可在单一芯片上包含百万颗晶体管。20年后的现今，64位、32-纳米装置已被量产，其在单一芯片上具有十亿颗晶体管，且包含多个微处理器核心(microprocessor core)来处理数据。

[0003] 除了指令平行应用在现今的多核心处理器(multi-core processor)，乱序执行(out-of-order execution)机制也被广泛的使用。根据乱序执行规则，指令以队列的方式储存在保留站(reservation station)以供执行单元来执行，且只有因为是旧指令(older instruction)的执行，而在等候操作数(operand)的那些指令，才会被拦截到保留站，没有在等候操作数的指令则会直接被派送去执行。接下来，执行的结果会被依适当的顺序以队列的方式排列并放回缓存器。传统上在处理器阶段(processor stage)，会被称为收回状态(retire state)。因此，指令并未依照原先程序的顺序来执行。

[0004] 因为除了在闲置的状态，执行单元可用以当旧的指令在等候其操作数时，执行新的指令(younger instruction)，因此乱序执行改善了庞大流量的问题。如同此领域具有通常知识者所了解的，指令不会总是成功地执行，因此当给定的指令无法成功地执行时，那个指令和其它新的指令就必须被重新执行。因为目前的机制，处理器会停止目前的执行，退回机器状态(machine state)至无法成功执行指令之前的时间点，且重新执行未成功执行的指令和在未成功执行的指令被派送前已派送或未派送的所有新的指令，因此这样的概念被称为“重新执行(replay)”。

[0005] 然而，重新执行是一例外情况(exceptional case)，且重新执行的性能影响通常是可忽略的。然而，缓存在保留站直到获得其操作数的缓存指令的性能影响则很大，微处理器的设计者已发展了加速技术，以允许当在执行前，有很高的可能性可取得指令的操作数时，特定的指令会先被派送。不仅特定的指令会被派送，运行的机制可适当地及时提供这些指令所需的操作数。

[0006] 在这应用中提出了一种加速技术，在此加速技术中，在其执行会导致从高速缓存查询操作数的加载指令被派送后，需要被假设有很高的机率存在在核心上的高速缓存的操作数的新的指令，会根据规定的数量的时钟周期被派送。因此，当加载指令被派送时，多个在等待操作数的新指令会被安置在各自的保留站中，直到所规定的数量的时钟周期结束，然后新指令会被派送，以进行具有高确定性的执行，也就是新指令将可取得其所需的操作数。

[0007] 使用上述所提到的加速技术所产生的性能改善是显著的，所以微处理器的设计者传统上会全面地应用此技术在所有加载指令(例如：来自输入/输出的加载、不可高速缓存的加载(uncacheable loads)、来自中断缓存器(interrupt register)的加载、特定加载(special loads)等)，尽管当有许多加载指令时确实会花费比 制订周期还要长的时间来取得其操作数，因此在预期可获得操作数的情况下，需要重新执行所有被派送的新指令。使用上述的加速技术所产生的性能改善大大补偿了，不常发生的重新执行所导致的性能损失。

[0008] 但是随着多核心处理器技术持续的进步，设计者发现了很少被存取的特定处理器资源，例如：层级2高速缓存(level 2(L2)cache)、中断控制器(interrupt controller)、熔丝阵列(fuse array)等，比较适合配置在多核心处理器芯片上共有的区域，而不是复制(replicate)到每一核心中。因此，像是上面所提到的资源，会被处理器核心所共享。如同此领域具有通常知识者所了解的，从不在核心的资源(off-core resource)(例如：熔丝阵列)中加载操作数所花费的时间会比从在核心的资源(on-core resource)(例如：层级1高速缓存(L1cache))中加载还要长。此外，纵使根据上述加速技术需要重新执行被派送的新指令所导致的性能损失并不大，但目前的发明人员发现到，其对电源使用的影响是值得注意的，对于庞大数量的指令在上述条件下被执行时，指令几乎都会被重新执行。此外，这些指令的起始执行(initial execution)本质上其实是浪费电源的，因此，这样的情况对于电池寿命(battery life)、热轮廓(thermal profile)和可靠度(reliability)而言是不利的。

[0009] 因此，能够经由改善重新执行的所需次数以节省处理器操作时所花费的电源的设备和方法是需要的。

[0010] 此外，在一乱序处理器中，为了节省处理器的电源，降低附载重新加载的机制是需要的。

发明内容

[0011] 有鉴于上述现有技术的问题、缺点和限制，本发明提供了用以改善在一处 理器重新执行加载的装置和方法。

[0012] 根据本发明的一实施例提供了一种用以改善在一乱序处理器重新执行加载的装置。上述装置包括了第一保留站以及第二保留站。所述第一保留站用以派送第一加载微指令，以及若所述第一加载微指令是指示多个非核心资源的其中一个的多个规定的加载微指令的其中一个，用以在保留总线进行侦测和指示。所述第二保留站耦接至所述保留总线，且在所述第一加载微指令派送后的第一数量的时钟周期之后，用以派送和所述第一加载微指令相依的一或多个新的微指令以进行执行，以及若在所述保留总线上指示了，所述第一加载微指令是所述多个规定的加载微指令的其中一个，所述第二保留站用以缓存所述一或多个新的微指令的派送，直到所述第一加载微指令取得所述操作数。所述多个非核心资源，包括：熔丝阵列。熔丝阵列用以储存对应所述乱序处理器的所述多个规定的加载微指令，其中在初始化时，所述乱序处理器存取所述熔丝阵列，以决定所述多个规定的加载微指令。

[0013] 根据本发明的另一实施例提供一种用以改善重新执行加载的装置。所述装置包括具有多个核心的多核心处理器。每一所述多个核心包括第一保留站，以及第二保留站。所述第一保留站用以派送第一加载微指令，以及若所述第一加载微指令是指示多个非核心资源

的其中一个的多个规定的加载微指令的其中一个，用以在保留总线进行侦测和指示。所述第二保留站耦接至所述保留总线，且在所述第一加载微指令派送后的第一数量的时钟周期之后，用以派送和所述第一加载微指令相依的一或多个新的微指令以进行执行，以及若在所述保留总线上指示了，所述第一加载微指令是所述多个规定的加载微指令的其中一个，所述第二保留站用以缓存所述一或多个新的微指令的派送，直到所述第一加载 微指令取得操作数。所述多个非核心资源，包括熔丝阵列。熔丝阵列用以储存对应所述乱序处理器的所述多个规定的加载微指令，其中在初始化时，所述乱序处理器存取所述熔丝阵列，以决定所述多个规定的加载微指令。

[0014] 根据本发明的另一实施例提供了一种用以改善在乱序处理器重新执行加载的方法。所述方法的步骤包括：配置多个非核心资源，其中所述多个非核心资源包括熔丝阵列，用以储存对应所述乱序处理器的所述多个规定的加载微指令；经由第一保留站派送第一加载微指令，以及若所述第一加载微指令是指示从多个非核心资源的其中一个的多个规定的加载微指令的其中一个，用以在保留总线进行侦测和指示；以及经由耦接至所述保留总线的第二保留站，在所述第一加载微指令派送后的第一数量的时钟周期之后，派送和所述第一加载微指令相依的一或多个新的微指令以进行执行，以及若在所述保留总线上指示了，所述第一加载微指令是所述多个规定的加载微指令的其中一个，所述第二保留站用以缓存所述一或多个新的微指令的派送，直到所述第一加载微指令取得所述操作数。

[0015] 关于本发明其它附加的特征与优点，此领域的熟习技术人士，在不脱离本发明的精神和范围内，当可根据本案实施方法中所揭露的执行联系程序的使用者装置、系统、以及方法，做些许的更动与润饰而得到。关于产业利用性，本发明可应用在一般目的或特别目的的运算装置中使用的微处理器。

附图说明

[0016] 图1为显示目前使用配置在每一核心101外部的共享资源的多核心处理器的方块图100；

[0017] 图2为显示在图1中每一当前的核心101中例示性的核层(core stage)的 方块图200；

[0018] 图3为显示根据本发明的实施例所述的对于来自非核心资源的加载具有一节能机制的多核心处理器的区块图300；

[0019] 图4为显示在图3中每一核心301中例示性的核层(core stage)的方块图400；

[0020] 图5为根据本发明的实施例所述的图4的非核心缓存组件461的区块图500；

[0021] 图6为根据本发明的实施例所述的图4的每一保留站RS1-RSN的方块图600；

[0022] 图7为根据本发明的实施例所述的图4的非核心未命中组件462的方块图700。

具体实施方式

[0023] 以下描述本发明的示例性和说明性的实施例。为了清楚起见，在该说明书中没有对实际实现的所有功能进行描述，对于本领域具有通常知识者而言将会理解的是，在对于任何该实际实现的开发中，可以进行许多实施方案特定的决定，以实现特定目标，诸如符合系统相关和业务相关的约束，其可以根据实现方案而不同。此外，应该理解的是，该开发工

作可能是复杂和耗时的，但是对于受益于本公开的本领域具有通常知识者而言其仍然是例行的任务。优选实施例的各种修改对于本领域具有通常知识者而言将是显而易见的，并且本文中所界定的一般原理可应用于其它实施例。因此，本发明并不限于这里示出和描述的具体实施例，而是应被赋予与这里所公开的原理和新颖特征相一致的最广范围。

[0024] 本发明将参考附图进行说明。在附图中示意性地示出的各种结构、系统、和设备仅仅是出于解释的目的，以便使其不以本领域具有通常知识者所公知的细节来模糊了本发明。然而，包括附图来描述和解释本发明的说明性示例。应该理解的是，在此所用的字汇和词组应该被理解和解释为具有与现有技术的本领域具有通常知识者对于这些字汇和词组的理解一致的含义。在此，术语或者短语的特定定义，(即不同于本领域具有通常知识者所理解的常规和习惯意思的定义)没有通过术语或者短语的一贯的使用来暗示。对于用以具有特定含义(即具有不同于技术人员所理解的含义)的术语或短语，将以直接和毫不含糊地提供用于对于术语或短语的特定定义的定义方式来在说明书中清楚地阐明该特定的定义。

[0025] 集成电路(Integrated Circuit,IC)：组装在半导体材料上的小部分区块(传统上是硅芯片(silicon))的电子电路集合。集成电路(IC)也可称为芯片(chip)、微芯片(microchip)，或晶元(die)。

[0026] 中央处理单元(Central Processing Unit,CPU)：通过对数据进行运算(包括算术运算(arithmetic operation)、逻辑运算(logic operation)，以及输入/输出运算(input/output operation))，来执行计算机程序(也称为“计算机应用”或“应用程序”)的指令的电子电路(即硬件)。

[0027] 微处理器(Microprocessor)：在一单一集成电路上具有如同中央处理单元功能的电子装置。微处理器接收数字数据作为输入，根据从存储器(在芯片上(on-die)或不在芯片上(off-die))撷取的指令来处理数据，以及将经由指令指示的操作产生的结果作为输出。一般目的的微处理器会使用在桌上型计算机(desktop)、移动电话(mobile)，或平板计算机(tablet computer)中，以及用来进行计算、文字编辑、多媒体播放，以及浏览网络。微处理器也会被配置在 嵌入式系统以控制不同的装置，这些包括电器用品(appliance)、移动电话、智能型手机(smart phone)，以及工业控制装置(industrial control device)。

[0028] 多核心处理器(Multi-Core Processor)：也可称作多核心微处理器(multi-core microprocessor)。多核心处理器表示具有配置在单一集成电路上的多个中央处理单元(核心)的微处理器。

[0029] 指令集架构(Instruction Set Architecture,ISA)或指令集(Instruction Set)：计算机结构中与程序设计有关的部分，包含了数据类型(data type)，指令集，缓存器，寻址模式(addressing mode)，存储器结构(memory architecture)，中断(interrupt)，异常处理(exception handling)以及输入/输出(input/output)。指令集架构包含运算码(opcode)(即机器语言指令)集合的设定，以及由特定中央处理单元执行的原生指令(native command)。

[0030] x86-兼容性微处理器(x86-Compatible Microprocessor)：能够执行根据x86指令集架构所转译的计算机应用的微处理器。

[0031] 微程序代码(Microcode)：用以进行与多个微指令(micro instruction)相关的项目。微指令(也称作原生指令(native instruction))，是在微处理器子单元(sub-unit)层

级的指令。例示性的子单元包括整数单元(integer unit)、浮点单元(floating point unit)、多媒体单元(multimedia(MMX)unit),以及加载/储存单元(load/store unit)。举例来说,会经由精简指令集计算机(reduced instruction set computer,RISC)微处理器来直接执行微指令。复杂指令集计算机(complex instruction set computer,CISC)微处理器,例如:x86-兼容性微处理器,x86指令会转译成相关的微指令,且相关的微指令会直接经由子单元或在复杂指令集计算机微处理器内的子单元来执行。

[0032] 熔丝(Fuse):导电结构,传统上会以灯丝(filament)来配置,其能够经由让电压通过灯丝和/或电流流过灯丝,在所选取的位置被断开。熔丝会经由习知的制作技术(fabrication technology)配置在芯片表面型态上的特定的区域,以在所有可能程序化的区域都配置灯丝。熔丝结构会熔断(或不熔断)在制作后,可提供配置在芯片上的对应装置所想要的可程序化特性。

[0033] 根据上述讨论到的多核心处理器中加载机制的背景,以及为了从非核心(non-core)资源执行加载,使用在目前多核心处理器的相关技术,关于目前技术的限制及缺点的讨论将通过图1-2来做说明。接者,会通过参考图3-7,来说明本发明的讨论内容。

[0034] 回到图1,图1为显示目前使用配置在每一核心101外部的共享资源的多核心处理器的方块图100。方块图100显示了装置芯片110,在装置芯片110上配置了四个处理器核心101。特别说明的是,在此所述的四核心(即四个核心101)的多核心微处理器仅是用以说明,但并非用以限制本发明。本发明所提出的原理和技术特征也可应用在具有不同数目的核心101的多核心微处理器。

[0035] 如同本领域具有通常知识者所了解的,为了设计和/或商业相关的原因,设计者会在核心101间选择共享某些处理器资源。以性能为考虑而言,这些共享资源传统上会如同核心101一样被配置在相同的装置芯片110上,且核心101会经由高速总线(high speed busses)111-114来存取这些共享资源。方块图100显示了例式性的共享资源,例如:层级2高速缓存103、随机存取存储器(random access memory,RAM)104、总线单元(bus unit)105、输入/输出单元106、高阶可编程中断控制器(Advanced Programmable Interrupt Controller,APIC)107,以及熔丝阵列108。整体而言,这些如同核心101配置在相同的装置芯片110上,但却是配置在在核心101外部的共享资源103-108,在之后内容将会视为非核心资源102。因此,总线B1 111使得核心1 101能够存取非核心资源102。总线B2 112使得核心2 101能够存取非核心资源102。总线B3 113使得核心3 101能够存取非核心资源102。总线B4 114使得核心4 101能够存取非核心资源102。在传统上的配置,多核心处理器会耦接至在装置芯片110外部的其它组件,例如:系统存储器(经由一存储器总线MEM进行存取)、输入/输出组件(经由总线I/O进行存取),以及系统控制组件(经由控制总线CTRL进行存取)。上述系统存储器包括一或多页表,用以储存虚拟地址以及实体地址间的一或多映像,并且可进行上述一或多页表的页表查找以撷取操作数。在其它实施例中,非核心资源、系统存储器、输入/输出组件或系统控制组件可能未和多核心处理器被安置在相同的芯片上,而通过总线彼此耦接。

[0036] 在操作上,每一核心101在操作系统控制下会执行从系统存储器所撷取的相关的指令,以及执行对应想要执行的应用的操作数。一或多个核心101可存取一或多非核心资源102,且将经由对应的总线B1-B4以控制的方式来存取一或多非核心资源102。举例来说,在

电源初始设定期间，一或多核心101会从熔丝阵列108执行加载操作以取得配置参数(熔丝阵列108储存对应的核心的配置数据、对应的加载微指令)，或会从随机存取存储器104执行加载操作以取得派送信息(例如是：处理器的微程序代码)。在一般操作时，核心101会存取层级2高速缓存103以读取/写入不在核心上高速缓存(例如层级1高速缓存)的存储器操作数。核心101会存取总线单元105，以进行读取/写入存储器系统的动作，或者核心101会存取输入/输出单元106以经由输入/输出总线执行输入/输出的操作。核心101还会存取高阶可编程中断控制器107以执行中断操作。

[0037] 现在来谈论图2，图2为显示在图1中每一当前的核心101中例示性的核层(core stage)的方块图200。在图2中显示了配置在装置芯片110的处理器核心201。处理器核心201包括经由总线241耦接至转译层(translator stage)212的提取层(fetch stage)211。转译层212经由总线242耦接至更名层(rename stage)213。更名层213经由总线243耦接至重新执行多工层(replay mux stage)214。重新执行多工层214经由总线244耦接至多个保留站RS1 221.1-RSN 221.N以及加载保留站RSL 221.L。每一保留站RS1 221.1-RSN 221.N、RSL 221.L经由对应的派送总线251.1-251.N、251.L，耦接至对应的执行单元EU1 222.1-EUN 222.N、EUL 222.L。保留站RS1 221.1-RSN 221.N、RSL 221.L经由总线245耦接至缓存数据226。每一保留站用以在所对应的总线(保留总线)进行侦测和指示。

[0038] 注意地是，除了执行单元EUL外，剩余的执行单元EU1-EUN会包括典型超纯量处理器所包含的单元，例如整数单元、浮点单元、多媒体单元以及储存单元。接下来则特别说明执行单元EUL，执行单元EUL其主要功能是从不同资源加载操作数，不同资源可像是系统存储器、系统输入/输出，以及如同图1所述的非核心资源230。

[0039] 此外，执行单元EUL通过总线254耦接至层级1高速缓存223，以及通过总线256耦接至非核心资源230。对于大部分存储器操作数而言，加载单元222.L会先存取层级1高速缓存223。若加载在层级1高速缓存223中未命中MISS，加载单元222.L就必须存取在非核心资源230的层级2高速缓存。执行单元EU1-EUN、EUL也经由总线252耦接至重排缓冲器(reorder buffer)224。此外，执行单元EUL通过对应未命中信号MISS的总线253耦接至重排缓冲器224。重排缓冲器224通过对应重新执行信号REPLAY的总线258耦接至重新执行多 工层214，且通过总线257耦接至收回单元(retire unit)225。收回单元通过对应写回信号WB(write back)的总线255耦接至缓存数据226。

[0040] 注意地是，在图2所示的核层中，处理器核心201仅用以举例说明现今的超纯量处理器，或乱序处理器核心，以及仅用以为了说明本发明，但本发明并不以此为限。对于此领域具有通常知识者而言，处理器核层可根据不同的架构以及应用而有所变化。

[0041] 在操作上，程序指令(未显示)是提取层(提取单元)211从存储器所提取。在一x86-兼容性微处理器核心201中，这些程序指令和x86指令集架构一致。程序指令依序在总线241上提供给转译层212。转译层212转译这些程序指令成一或多个微指令，这些微指令用以在对应的执行单元EU1-EUN、EUL中来指示次操作(sub-operation)，以执行程序指令所指示的操作。接着微指令会在总线242上提供给更名层213，其中在一些微指令所指示的结构缓存器(architectural register)(即操作数缓存位置)会重新映像至在处理器核心201中的硬件缓存器(未显示)，以加速对于独立微指令串(micro instruction stream)的执行平行化(execution parallelism)。更名层213也会根据串行的程序顺序(serial program order)

来标记(tag)每一微指令，且在微指令中的资源和目标字段也会被标记为和这些微指令的一或多操作数所相依的新微指令的卷标。接着，更名的微指令会在总线243上提供给重新执行多工层214。

[0042] 重新执行多工层214执行在乱序处理器核心201中的多个指令。首先，重新执行多工层214会读取在每一更名的微指令中的操作数，以决定适当的执行单元EU1-EUN、EUL来进行执行。举例来说，更名的整数微指令会通过执行单元EU1来执行。浮点微指令会通过执行单元EU2来执行等等。此外，在目前应 用特别关注地是，更名的加载微指令会通过执行单元EUL来执行。因此，重新执行多工层214会提供一或多个更名的微指令至一或多保留站RS1-RSN、RSL，以让这些指令在保留站RS1-RSN、RSL等候被派送至对应的执行单元EU1-EUN、EUL。

[0043] 每一保留站RS1-RSN、RSL存取缓存数据226以读取队列排序用以进行操作的更名的微指令所需的操作数。未标示为旧的更名的微指令的卷标的更名的微指令(即此更名的微指令并未和旧的更名的微指令相依(dependent))会立即被派送至对应的执行单元EU1-EUN、EUL进行执行。相依的更名的微指令(即包含有尚未完成执行的旧的更名的微指令的卷标的更名的微指令)会逐一的储存到保留站RS1-RSN、RSL中，直到可获得其被标示的相依的操作数时为止。当可获得被标示的相依的操作数时，这些操作数会提供给其所相依的更名的微指令，且这些更名的微指令会被派送至对应的执行单元EU1-EUN、EUL进行执行。当执行单元EU1-EUN、EUL不执行微指令时，其会执行节能功能。一般来说，当执行单元EU1-EUN、EUL不执行微指令时，在执行单元EU1-EUN、EUL内的时钟(clock)会停止以节省大量的电源。

[0044] 更名的微指令以及其对应的相关结果会经由总线252提供给重排缓冲器224。重排缓冲器224会安置来自更名的微指令的乱序执行所产生的所有结果，恢复到原来的程序顺序。也就是说，来自更名程序缓存器的结果会重新映像至其所对应的结构缓存器(architectural register)，且会根据执行的特定程序顺序，将这些结果排序进入结构缓存器中。已经成功完成执行具有适当结果的微指令会在总线257上提供给收回单元225。收回的微指令的结果会在对应写回信号WB的总线255上被写回至缓存数据226中。

[0045] 如同此领域具有通常知识者所了解的，有许多情况会造成更名的微指令无法成功地执行，举例来说(但不以此为限)，像是程序例外(program exception)，一般中断(general interrupt)、输入/输出中断(I/O interrupt)、分支例外(branch exception)等等。当重排缓冲器224判断更名的微指令无法成功地执行时，其必须和已发送用以执行的所有新的更名的微指令一起被重新发送(重新执行)。因此，重排缓冲器224会在对应重新执行信号REPLAY的总线258上，通过提供无法成功被执行的更名的微指令的卷标，启始重新执行事件。

[0046] 当无法成功被执行的更名的微指令的卷标被提供给重新执行多工层214时，重新执行多工层214会回应一机器状态，以和更名的微指令的执行相一致，其中此更名的微指令的执行起始于其卷标在总线258上被提供的更名的微指令。

[0047] 如同此领域具有通常知识者所了解，为了改善性能，微处理器的设计者经常对关于程序将如何执行作出假设。举例来说，在此领域中所熟知的作法，不会采用具有庞大比例的分支。因此，提取单元(提取层)211会根据这样的假设用以让指令排队等候执行。若一分支未被采用，整体的执行速度就会被改善。若此分支被采用，所有比此分支还旧的指令必须被来自所采用的程序路径的指令所取代。

[0048] 微处理器的设计所做的另一假设为加载微指令将会在特定数目的时钟周期中，在层级1高速缓存223命中(hit)，且此假设是基于根据存取层级1高速缓存223的设计，在所需时钟周期数目内，在层级1高速缓存223命中的统计量，例如：百分之90的命中率。本发明中会采用4时钟周期来存取层级1高速缓存223，然而，这仅是用以作为说明的目的，本发明并不以此为限。其它时钟周期的数目也可被考虑。

[0049] 此外，保留站RS1-RSN可包括逻辑组件。在派送旧的加载指令之后，逻辑组件用以缓存(stall)其标签是对应旧的加载微指令的更名的微指令，直到过了四个时钟周期。然后，假设在四个时钟周期期间，旧的微止令将会在层级1高速缓存223命中，且将可取得标示的操作数的情况下，派送新的更名的微指令至其所对应的执行单元EU1-EUN。虽然在图2未显示，但必须注意地是，执行单元EU1-EUN、EUL也可存取在加载操作可获得的操作数，且提供这些操作数给目前执行的微指令。对于层级1高速缓存223命中的加载而言，操作数会被提供给所派送的相依的新的微指令，其中此新的微指令执行完成的速度会快于提供原先的操作数的情况。但是对于无法在层级1高速缓存223中命中的加载(根据百分之90的命中率的假设下，未命中的机率大概为百分之10)而言，在加载成功完成后，在要命中的假设下，被派送的所有相依的新的微指令，都必须被重新执行。因此，当无法在层级1高速缓存223命中时，执行单元EUL将会通过在对应未命中信号MISS的总线253上指示未命中的加载指令的卷标，来通知重排缓冲器224，以启动新的相依的指令的重新执行。

[0050] 以性能的观点来看这样的方案是极有效的，对大多数目前的高速缓存是非常有效率的，在一般施行的方式会在加载指令派送后，根据假定的高速缓存存取数目的时钟周期(例如：四个时钟周期)，在此数目的时钟周期，缓存与加载微指令相依的所有微指令。相依的微指令会停留在各自的保留站RS1-RSN中，且当假设规定在加载微指令的操作数可从层级1高速缓存223获得时，相依的微指令就会被派送。一般来说，此加速方法会使用在所有加载指令，这些加载指令包括了用以存取除了层级1高速缓存223之外的资源的加载指令。因为这样类型的加载指令比起存储器加载指令是相对地不常见，所以关于重新执行存 取除了存储器之外的资源的加载指令对性能所产生的影响一般而言是可被忍受的。举例来说，当取得加载微指令超过特定数量的时钟周期(例如：四个时钟周期)才成功执行(即解决(resolve))，执行单元EUL将会表示有不命中在对应未命中信号的总线，因此造成了在完成加载后，相依的新的微指令会被重新执行。

[0051] 上述技术在过去几年良好地运用在改善超纯量处理器201的性能，然而当这样的方法被应用在如图1所示的多核心处理器的配置时，现今的发明人员发现到了会有额外的挑战因此产生出来。更具体地来说，像这样的方法虽然通过存取到层级1高速缓存223所控制的配置中相当有效，但在电源的使用上并没有效率，当此方法应用在越来越频繁用以存取非核心资源230的多核心处理器配置时，和使用在目前层级1高速缓存223相比，对于非核心资源230的存取时间是非常慢的。

[0052] 也就是说，任何用以特别指示非核心资源(规定资源)230(例如：熔丝阵列108、总线单元105(针对不能高速缓存的加载)、高阶可编程中断控制器107、输入/输出单元106、层级2高速缓存103、以及随机存取存储器104)的加载微指令，将导致具有来自非核心资源230的加载的标签的相依的新的微指令重新执行。此外，目前发明人员观察到，当没有太多的性能命中(performance hit)，这些相依的新微指令所浪费的起始执行(initial execution)

会导致庞大的电源花费，由于保证被重新执行的加载微指令会被派送至执行单元EU1-EUN，因此将会使用到原本会通过内部电源管理机制节省下来的电源。

[0053] 本发明根据一新的方法，提供了通过降低加载重新执行的次数，以节省乱序多核心处理器中的电源的一设备和方法，来克服了上述所关注的目前的加载 机制的限制。此新的方法将会在底下参考图3-7来进行谈论。

[0054] 现在来参考图3。图3为显示根据本发明的实施例所述的对于来自非核心资源的加载具有节能机制的多核心处理器的区块图300。区块图300显示了装置芯片310，在装置芯片310配置了四个处理器核心301。特别说明的是，在此所述的四核心301仅是用以说明，但并非用以限制本发明。本发明所提出的原理和技术特征也可应用在具有不同数目的核心301，其中来自非核心资源的加载操作所花费的存取时间，会长于在核心上高速缓存的存取。

[0055] 如同图1所示的多核心处理器，根据本发明的实施例所述的多核心处理器会包括非核心资源302，非核心资源302传统上会如同核心301一样配置在相同的芯片310上，且每一核心301会经由高速总线311-314来存取非核心资源302。区块图300显示了例示性的共享资源，例如：层级2高速缓存303、随机存取存储器304、总线单元305、输入/输出单元306、高阶可编程中断控制器307，以及熔丝阵列308，但本发明并不以此为限。因此，总线B1 311使得核心1 301能够存取非核心资源302。总线B2 312使得核心2 301能够存取非核心资源302。总线B3 313使得核心3 301能够存取非核心资源302。总线B4 314使得核心4 301能够存取非核心资源302。在传统上的配置，多核心处理器会耦接至在芯片310外部的其它组件，例如：系统存储器(经由存储器总线MEM进行存取)、输入/输出组件(经由总线I/O进行存取)，以及系统控制组件(经由控制总线CTRL进行存取)，但本发明并不以此为限。上述系统存储器包括一或多页表，用以储存虚拟地址以及实体地址间的一或多映像，并且可进行上述一或多页表的页表查找以撷取操作数。在其它实施例中，非核心资源、系统存储器、输入/输出组件或系统控制组件可能未和多核心处理器被安置在相同的芯片上，而通过总线 彼此耦接。系统控制组件可包括，快速外围组件互连(Peripheral Component Interconnect Express，PCI-e)组件、外围组件互连(Peripheral Component Interconnect，PCI)组件、通用序列总线(Universal Serial Bus，USB)组件、图形适配器(Graphics Adaptor)、共处理器(co-processor)，以及处理器间(inter-processor)沟通组件，但并不以此为限。

[0056] 和图1的多核心处理器相比较，本发明所述的多核心处理器在每一核心301中包括了重新执行缩减器(replay reducer)组件320。在一实施例中，重新执行缩减器组件320用以侦测除了在核心高速缓存(未显示)外用以指示资源的加载，以缓存派送的新的相依微指令直到加载结束，以及排除原本会导致重新执行的任何指示的判定。因此，在核心301的一或多执行单元(未显示)，由于缓存了新的相依微指令的派送的关系，会进入电源管理模式，也因此可以节省芯片310上原本会浪费的电源。

[0057] 在操作上，每一核心301在操作系统的控制下，会执行从系统存储器中所撷取的相关的指令，且会执行对应预期应用的操作数上的操作。一或多个核心301会需要存取一或多非核心资源302，且会经由对应的总线B1-B4，透过控制方法，存取一或多非核心资源302。举例来说，在电源初始化期间，一或多核心301会从熔丝阵列308执行加载操作，以撷取配置参数(熔丝阵列308储存对应的核心的配置数据、对应的加载微指令)，或者会从随机存取存储器304执行执行加载操作，以撷取微码插入(micro patch)和/或其它配置信息(例如是：处

理器的微程序代码)。在一般操作期间,核心301会存取层级2高速缓存303,以读取/写入已从系统存储器高速缓存的存储器操作数,其中此存储器操作数并不会出现在核心的高速缓存中(例如:层级1高速缓存)。核心301会存取总线单 元305,以读取/写入系统存储器,或者核心301会使用输入/输出单元306以经由输入/输出总线执行输入/输出的操作。核心301会存取总线单元305,以从控制组件读取控制数据,或写入控制数据至控制组件。核心301还会存取高阶可编程中断控制器307,以执行中断操作。

[0058] 并非自动地表示由于来自非核心资源302的加载的不命中,导致相依的新的微指令串的重新执行,而是重新执行缩减器组件320会缓存(延迟)相依的新的微指令串的执行,直到加载结束,因此使得执行单元的电源管理功能能够被使用。在一实施例中,重新执行缩减器组件320也会侦测未明确针对非核心资源308但原先被保证会造成未命中指示的其它类型的加载。其它类型的加载包括,输入/输出加载、需要特定数量的周期的加载、习知的需要页表查找(page table walk),以进行像是关于第二层级地址转译的加载(即巢状呼叫(nested paging)、x86扩展页面表(extended page table)加载),由x86特定总线周期(例如:关闭(shutdown)、暂停(halt)、清除(flush)等)的执行所产生的加载,以及已知解决无法高速缓存空间的加载、或已知解决写入结合的存储器空间的加载。其中,上述的加载指令用以决定系统存储器的写入结合的存储器空间。在其它实施例中则考虑了任何类型的加载操作的侦测,其中这样的做法有很大的可能性会花费比规定的数量的时钟周期还长的时间来完成。

[0059] 回到图4,图4为显示在图3中每一核心301中例示性的核层(core stage)的方块图400。在方块图400中显示了配置在芯片310的处理器核心401。处理器核心401包括经由总线441耦接至转译层(translator stage)412的提取层(fetch stage)411。转译层412经由总线442耦接至更名层(rename stage)413。更名层413经由总线442耦接至重新执行多工层(replay mux stage)414。

[0060] 重新执行多工层414经由对应保留和保留信号HOLDY的总线244耦接至多个保留站RS1 421.1-RSN 421.N以及增强加载保留站ERSL 421.L。增强加载保留站ERSL 421.L包含非核心缓存组件461。每一保留站RS1-RSN、ERSL经由对应的派送总线451.1-451.N、451.L,耦接至对应的执行单元EU1 422.1-EUN 422.N、EUL 422.L。保留站RS1-RSN、ERSL经由总线445耦接至缓存数据426。每一保留站用以在所对应的总线(保留总线)进行侦测和指示。

[0061] 注意地是,除了执行单元EUL外,剩余的执行单元EU1-EUN会包括典型超纯量处理器所包含的单元,例如整数单元、浮点单元、多媒体单元以及储存单元。执行单元EUL是加载单元422.L,且其主要功能是从不同资源加载操作数,不同资源可像是系统存储器、系统输入/输出,以及如同图3所述的非核心资源430,但本发明并不以此为限。执行单元EUL还包括一核心未命中组件(uncore miss)UMISS 462。

[0062] 此外,执行单元EUL通过总线454耦接至层级1高速缓存423,以及通过总线456耦接至非核心资源230。对于存储器操作数而言,加载单元(执行单元)422.L会先存取层级1高速缓存423。若加载在层级1高速缓存423中未命中MISS,加载单元422.L就必须存取在非核心资源430的层级2高速缓存(未显示)。执行单元EU1-EUN、EUL也经由总线452耦接至一重排缓冲器(reorder buffer)424。此外,执行单元EUL通过对应未命中信号MISS的总线453耦接至重排缓冲器424。重排缓冲器424通过用来对应重新执行信号REPLAY的总线458耦接至重新

执行多工层414，通过总线457耦接至收回单元(retire unit)425，以及通过对应保留和保留信号HOLDY的总线444耦接至保留站RS1-RSN、ERSL。收回单元通过对应写回信号WB(write back)的总线455耦接至缓存 数据426。

[0063] 注意地是，在图4所示的核层仅用以举例说明本发明，但本发明并不以此为限。因为图4所示的核层仅以乱序处理器核心来做为例子，对于此领域具有通常知识者而言，能够将本发明所提出的作法，应用在不同的架构以及应用所需的其它处理器核层的配置上。

[0064] 在操作上，程序指令(未显示)通过提取层(提取单元)411从存储器所提取。在一x86-兼容性微处理器核心401中，这些程序指令和x86指令集架构一致。程序指令依序在总线441上提供给转译层412。转译层412转译这些程序指令成一或多个微指令，这些微指令用以在对应的执行单元EU1-EUN、EUL中来指示次操作(sub-operation)，以执行程序指令所指示的操作。接着微指令会在总线442上提供给更名层413，其中在一些微指令所指示的结构缓存器(architectural register)(即操作数缓存位置)会重新映像至在处理器核心401中的硬件缓存器(未显示)，以加速对于独立微指令串(micro instruction stream)的执行平行化(execution parallelism)。更名层413也会根据串行的程序顺序(serial program order)来标记(tag)每一微指令，且在微指令中的资源和目标字段也会被标记为和这些微指令的一或多操作数所相依的新微指令的卷标。接着，更名的微指令会在总线443上提供给重新执行多工层414。

[0065] 重新执行多工层414会读取在每一更名的微指令中的操作数，以决定适当的执行单元EU1-EUN、EUL来进行执行，特别说明的是，更名的加载微指令会通过加载的执行单元EUL来执行。因此，重新执行多工层414会提供一或多个更名的微指令至一或多保留站RS1-RSN、ERSL，以让这些指令在保留站RS1-RSN、ERSL等候被派送至对应的执行单元EU1-EUN、EUL。

[0066] 每一保留站RS1-RSN、ERSL存取缓存数据426以读取队列排序用以进行操作的更名的微指令所需的操作数。未标示为旧的更名的微指令的卷标的更名的微指令(即此更名的微指令并未和旧的更名的微指令相依(dependent))会立即被派送至对应的执行单元EU1-EUN、EUL进行执行。除了将在底下谈论的内容，相依的更名的微指令(即包含有尚未完成执行的旧的更名的微指令的卷标的更名的微指令)会逐一的储存到保留站RS1-RSN、ERSL中，直到可获得其被标示的相依的操作数时为止。当可获得被标示的相依的操作数时，这些操作数会提供给其所相依的新的更名的微指令，且这些新的更名的微指令会被派送至对应的执行单元EU1-EUN、EUL进行执行。当执行单元EU1-EUN、EUL不执行微指令时，其可能会执行节能功能。当执行单元EU1-EUN、EUL不执行微指令时，在执行单元EU1-EUN、EUL内的时钟(clock)会停止以节省大量的电源。

[0067] 更名的微指令以及其对应的相关结果会经由总线452提供给重排缓冲器424。重排缓冲器424会安置来自更名的微指令的乱序执行所产生的所有结果，恢复到原来的程序顺序。也就是说，来自更名程序缓存器的结果会重新映像至其所对应的结构缓存器(architectural register)，且会根据执行的特定程序顺序，将这些结果排序进入结构缓存器中。已经成功完成执行具有适当结果的微指令会在总线457上提供给收回单元425。收回的微指令的结果会在对应写回的总线455上被写回至缓存数据426中。

[0068] 当重排缓冲器424判断更名的微指令无法成功地执行时，其必须和已发送用以执

行的所有新的更名的微指令一起被重新发送(重新执行)。因此,重排缓冲器424会在对应重新执行信号REPLAY的总线458上,通过提供无法成功被 执行的更名的微指令的卷标,启始重新执行事件。

[0069] 当无法成功被执行的更名的微指令的卷标被提供给重新执行多工层414时,重新执行多工层414会回应一机器状态,以和更名的微指令的执行相一致,其中此更名的微指令的执行起始于其卷标在总线458上被提供的更名的微指令。

[0070] 除了以下将会谈论的内容外,本发明包括了用以缓存微指令的保留站RS1-RSN,其中在规定的数量的时钟周期,这些被缓存的微指令和新加载微指令相依,且在加载微指令被派送后,假设此加载微指令将会在规定的数量的时钟周期,命中层级1高速缓存。在一实施例中,规定的数量的时钟周期为4个时钟周期。其它数量的时钟周期也可被考虑。

[0071] 此外,除了以下将会谈论的内容外,保留站RS1-RSN可包括逻辑组件。在派送旧的加载指令之后,逻辑组件用以缓存(stall)其标签是对应旧的加载微指令的更名的微指令,直到过了四个时钟周期。然后,假设在四个时钟周期,旧的微指令将会在层级1高速缓存423命中,且将可取得标示的操作数的情况下,派送新的更名的微指令至其所对应的执行单元EU1-EUN。虽然在图4未显示,但必须注意地是,执行单元EU1-EUN、EUL也可存取在加载操作可获得的操作数,且提供这些操作数给目前执行的微指令。对于在少于所规定的数量的时钟周期完成的加载而言,例如:在层级1高速缓存423命中的加载,操作数会被提供给所派送的相依的新的微指令,其中此新的微指令执行完成的速度会快于提供原先的操作数的情况。对于在超过所规定的数量的时钟周期完成的加载而言,例如:在层级1高速缓存423未命中的加载,在加载成功完成后,在要命中的假设下,被派送的所有相依的新的微指令,都必须被重新执行。因此,当无法在层级1高速缓存423命中时,执行单元EUL将会通过在总线453上指示 未命中的加载指令的卷标,来通知重排缓冲器424,以启动新的相依的指令的重新执行。

[0072] 然而,在本发明中也提供了有别于上述加速方法的作法,也就是在增强加载保留站ERSL 421.L中包含非核心缓存组件461。非核心缓存组件461用以侦测一或多加载微指令类型,以通过缓存新的微指令直到其取得算元,在一或多执行单元EU1-EUN中执行节能机制,其中新的微指令和一或多加载微指令类型的微指令相依。非核心未命中组件462(UMISS)也会侦测一或多加载微指令类型,以在当一或多加载微指令类型花费超过规定的数量的时钟周期来取得其操作数时,排除在对应未命中信号MISS的总线上的不命中。通过此做法,一或多加载微指令类型的微指令被允许完成执行,且因为新的相依的微指令已经在保留站RS1-RSN中缓存,所以不需要重新执行和一或多微指令类型的微指令相依的新的微指令。在本发明一实施例中,保留站RS1-RSN、ERSL会彼此进行关于侦测的一或多微指令类型的微指令的信息的沟通,或通过对应保留和保留信号HOLDY的总线444和重排缓冲器424进行关于侦测的一或多微指令类型的微指令的信息的沟通。当一或多加载微指令类型的微指令完成执行时,重排缓冲器42通过提供已经在对应保留和保留信号HOLDY总线444上完成执行的一或多加载微指令类型的微指令的卷标,来指示保留站RS1-RSN释出(release)其所缓存新的相依的微指令以进行派送。

[0073] 就优点而言,本发明的作法,在关于对应由系统存储器高速缓存的操作数的加载微指令上,提供了更有效的性能,以及大幅降低关于一或多微指令类型的微指令重新执行

的次数，因此，使得执行单元EU1-EUN在因为本发明缓存相依的微指令的应用而变成空闲(empty)时，能够进入节能模式(power saving mode)。

[0074] 因此，举例来说，特别用以指示非核心资源(规定资源)430的加载微指令，将不会导致具有来自非核心资源的加载的标签的新的相依的微指令重新执行，其中非核心资源430可像是熔丝阵列308、总线单元305、高阶可编程中断控制器307、输入/输出单元306、层级2高速缓存303、以及随机存取存储器304。

[0075] 在本发明一实施例中，一或多加载微指令类型会包括来自非核心资源430所规定的加载，其它类型的加载包括，输入/输出加载、需要特定数量的周期的加载、已知的需要页表查找(page table walk)的来自存储器系统的加载、由于x86特定总线周期(例如：关闭(shutdown)、暂停(halt)、清除(flush)等)的执行所产生的加载、已知解决无法高速缓存空间的加载、或已知解决写入结合的存储器空间的加载。在其它实施例中则考虑了任何类型的加载操作的侦测，其中这样的做法有很大的可能性会花费比规定的数量的时钟周期还长的时间来完成。

[0076] 在本发明一实施例中，非核心缓存组件461以及非核心未命中组件462(UMISS)会根据本发明所描述的侦测加载微指令类型的作法，用以执行处理器核心401的初始化(开启电源或重新设定)。本发明所描述的加载微指令类型会从熔丝阵列308特定的位置中被读取，以进行初始化。在本发明另一实施例中，每一处理器核心401会用以经由熔丝阵列308中的程序设计来侦测不同类型的规定的加载微指令，其中关于每一处理器核心401的微指令的类型会被设在熔丝阵列308中所对应的位置。更在本发明一实施例中，当经由联合测试工作群组(Joint Test Action Group，JTAG)接口(未显示)，在装置芯片310开启电源或重新设定时，规定的加载微指令的类型会被设计在随机存取存储器304 中，其中规定的加载微指令的类型会根据来自随机存取存储器304的特定区域的后续的初始化的结果来被读取。

[0077] 现在来谈论图5，图5为根据本发明的实施例所述的图4的非核心缓存组件461的区块图500。非核心缓存组件461包括微指令缓存器510，其中微指令缓存器510耦接至非核心加载操作数侦测逻辑电路501。微指令缓存器510包括微指令卷标字段OP TAG 511、运算码字段MICRO OP 512、资源A字段SRC A 513、标签A字段TAG A 514、资源B字段SRC B 515、标签B字段TAG B 516、资源C字段SRC C 517、以及标签C字段TAG C 518。非核心加载操作数侦测逻辑电路501产生保留信号HOLDY，且侦测逻辑电路501耦接至总线444。

[0078] 如同本领域具有通常知识者所知，目前的指令集架构(ISA)，例如：x86指令集架构，提供了许多不同的操作数寻址模式(operand addressing mode)，操作数寻址模式包括，直接寻址(direct)、间接寻址(indirect)、立即寻址(immediate)，以及相对寻址(relative)，但本发明并不以此为限。因此，资源字段SRC A-SRC C的一或多个会包括，操作数、操作数的一或多规定的位置(包括结果的目标)。因此，为了允许扩张本发明的应用，以应用在更多数量的指令集上，非核心缓存组件461的操作数将以像是资源字段SRC A-SRC C的一般认知的内容来做说明。

[0079] 在操作上，如同重新执行多工层414所提供的微指令，加载微指令会被输入至微指令缓存器510。微指令卷标字段OP TAG 511具有目前在微指令缓存器510的微指令的卷标。运算码字段MICRO OP 512则具有目前在微指令缓存器510的微指令的操作数。标签A字段TAG A 514的内容包括和资源A(SRC A)所相依的旧的微指令的卷标。标签B字段TAG A 516

的内容包括和资源B(SRC B)所相依的旧的微指令的卷标。标签C字段TAG C 518的内容包括和资源C(SRC C)所相依的旧的微指令的卷标。非核心加载操作数侦测逻辑电路501用以读取运算码字段MICRO OP 512的内容。若运算码字段MICRO OP 512不包括规定的加载操作数的其中一个,非核心加载操作数侦测逻辑电路501会取消设置(deassert)保留信号HOLDY,以指示保留站RS1-RSN,缓存在其内部的新的微指令会在适当的时候被派送,其中规定的加载操作数表示,上述会造成在和微指令缓存器510中目前的微指令相依的其它保留站RS1-RSN的新的微指令被缓存的操作数。然而,若运算码字段MICRO OP 512包括规定的加载操作数的其中一个,非核心加载操作数侦测逻辑电路501会设置(assert)保留信号HOLDY以及在总线444上安置微指令卷标字段OP TAG 511的内容,以指示保留站RS1-RSN,其内部的新的微指令必须被缓存直到微指令缓存器510中目前的微指令所规定的加载完成,以及加载的结果被提供给新的相依的微指令为止,其中规定的加载操作数表示,上述会造成在和微指令缓存器510中目前的微指令相依的其它保留站RS1-RSN的新的微指令被缓存的操作数。当加载完成时,重排缓冲器424将取消设置保留信号HOLDY,并释放缓存的微指令。

[0080] 回到图6,图6为根据本发明的实施例所述的图4的每一保留站RS1-RSN的方块图600。每一保留站RS1-RSN包括微指令缓存器610,其中微指令缓存器610耦接至相依性检查逻辑单元601。微指令缓存器610包括微指令卷标字段OP TAG 611、运算码字段MICRO OP 612、资源A字段SRC A 613、标签A字段TAG A 614、资源B字段SRC B 615、标签B字段TAG B 616、资源C字段SRC C 617、以及标签C字段TAG C 618。相依性检查逻辑单元601产生备妥信号READY,且监测耦接至总线444的逻辑电路501所产生的保留信号HOLDY。

[0081] 微指令缓存器610的字段611-618的内容和图5中具有相同名称的字段相同,因此可参考图5所述的内容。相依性检查逻辑单元601用以读取资源的标签字段TAG A-TAG C的内容。若任何标签字段TAG A-TAG C的内容和设置在保留信号HOLDY上的卷标相符,在微指令缓存器610中的微指令会被缓存,直到和在微指令缓存器610中的微指令相依的加载完成为止,此微指令经由加载取得的操作数,会被提供给对应的资源字段SRC A-SRC C,且重排缓冲器424取消设置保留信号HOLDY。当保留信号HOLDY被取消设置,相依性检查逻辑单元601设置备妥信号READY,以指示在微指令缓存器610中的微指令已准备好被派送至其所对应的执行单元EU1-EUN。

[0082] 若卷标字段TAG A-TAG C的内容和设置在保留信号HOLDY上的卷标不相符,相依性检查逻辑单元601会设置备妥信号READY,以指示在微指令缓存器610中的微指令已准备好被派送至其所对应的执行单元EU1-EUN。

[0083] 现在来参考图7,图7为根据本发明的实施例所述的图4的非核心未命中组件462的方块图700。非核心未命中组件462包括微指令缓存器710,其中微指令缓存器710耦接至加载未命中排除逻辑电路701。微指令缓存器710包括微指令卷标字段OP TAG 711、运算码字段MICRO OP 712、资源A字段SRC A 713、标签A字段TAG A 714、资源B字段SRC B 715、标签B字段TAG B 716、资源C字段SRC C 717、以及标签C字段TAG C 718。加载未命中排除逻辑电路701产生无未命中信号NOMISS(no miss signal)。

[0084] 微指令缓存器710的字段711-718的内容和图5-6中具有相同名称的字段相同,因此可参考图5-6所述的内容。加载未命中排除逻辑电路701用以读取运算码字段MICRO OP 712的内容。若运算码字段MICRO OP 712不包括规定 的上述会造成新的微指令被缓存的操

作数的其中一个，加载未命中排除逻辑电路701会取消设置无未命中信号NOMISS，以通知对应的加载执行单元EUL 422.L根据正常加载指令执行协议(normal load instruction execution protocol)管理无未命中信号NOMISS的状态。若运算码字段MICRO OP 712包括规定的操作数的其中一个，加载未命中排除逻辑电路701会设置无未命中信号NOMISS，以通知对应的加载执行单元EUL 422.L在微指令缓存器710的微指令的执行期间排除无未命中信号NOMISS的设置。

[0085] 本发明上述的组件用以执行如同本发明中所谈论的功能和操作。本发明所述的组件包括逻辑门、电路、装置或微程序代码(即微指令或客户指令(native instruction))，或逻辑门、电路、装置或微程序代码的组合，或用以执行如同本发明中所谈论的功能和操作的等效的组件。用以完成这些操作和功能的组件，会和在多核心处理器中用以执行其它功能和/或操作的其它电路或微程序代码等所共享。

[0086] 本发明的各部分和相应的详细描述利用软件，或对计算机内存内的数据位的操作的算法和符号表示予以呈现。这些描述和表示是本领域具有通常知识者借助其向本领域其它普通技术人员有效地传达其工作的实质的描述和表示。如在此所使用的算法(如其通常被使用的那样)被设想为是导致希望结果的前后一致的步骤序列。这些步骤是物理量的所需的物理操作。虽然未必需要，这些量通常采用能够被存储、传递、组合、比较以及另外被操作的光、电，或磁的信号的形式。为了一般使用的原因，将这些信号视为位、数值、元素、符号、字符、术语、数等是便利的。

[0087] 应该牢记的是，上述所有或者类似的术语与适当的物理量相关，并且其仅 仅用于方便标记所应用到的这些量。除非特别声明，或者是从讨论中可知，诸如“处理”，“计算”，“计划”，“确定”，“显示”或类似术语是指计算机系统、微处理器、中央处理单元、或类似的电子计算装置的动作或者处理，其将在计算机系统的寄存器和存储器内的表示物理、电子量的数据操纵和转换为相似地表示在计算机系统存储器或者寄存器，或者其它该种信息存储器、传输或者显示设备内的物理量的其它数据。

[0088] 还要注意的是，在本发明的软件实现方面，其被典型地编码在某种形式的程序存储介质上，或者被实施在几种类型的传输介质上。程序存储介质可以是电子的(例如，只读存储器、快存只读存储器、电可编程只读存储器)、磁随机存取存储器(例如，软盘或硬盘驱动器)、或光学器(例如，光盘只读存储器、或CD-ROM)，并且其可以是只读的或者是随机存取的。同样，传输介质可以是金属迹线、双绞线、同轴线缆、光纤、或本领域已知的一些其它合适的传输介质。本发明并不限于由任何给定的实施方式的这些方面。

[0089] 上述的具体公开的实施例仅仅是说明性的，对于本领域技术人员将会理解的是，可以使用所公开的概念和特定的实施例来作为基础而设计或者修改出用于执行与本发明相同的目的的其它结构，并且可以在不脱离由所附的权利要求书所阐述的本发明的范围的情况下，对本发明进行各种修改、替代、以及替换。

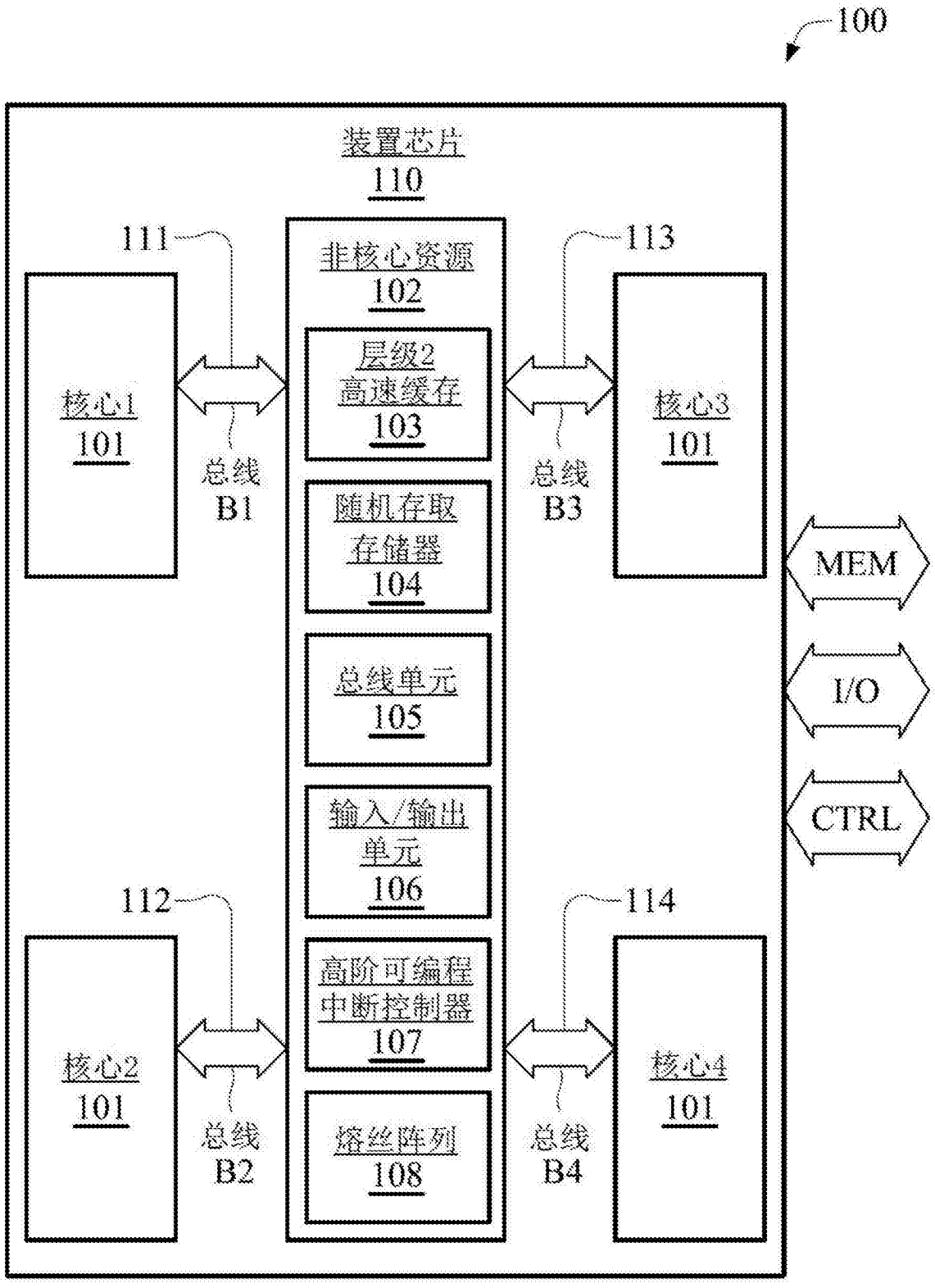

100
装置芯片
110
111
非核心资源
102
113
核心1
101
层级2
高速缓存
103
核心3
101
总线
B1
总线
B3
随机存取
存储器
104
MEM
总线单元
105
I/O
输入/输出
单元
106
CTRL
112
114
高阶可编程
中断控制器
107
核心2
101
核心4
101
总线
B2
熔丝阵列
108
总线
B4

图1

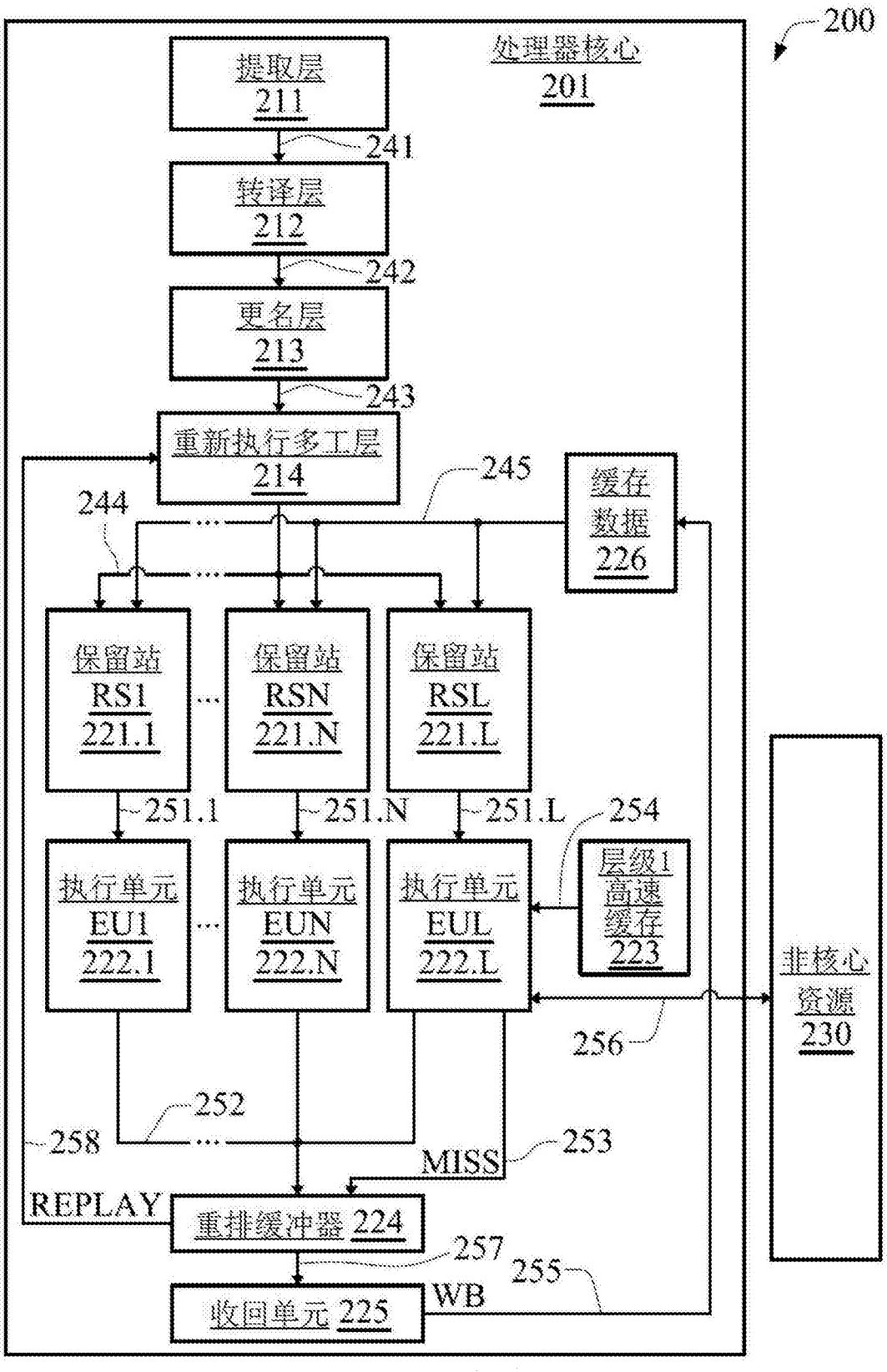

200
处理器核心
201
提取层
211
241
转译层
212
242
更名层
213
243
重新执行多工层
214
245
缓存
数据
226
244
保留站
RS1
221.1
保留站
RSN
221.N
保留站
RSL
221.L
251.1
251.N
251.L
254
执行单元
EU1
222.1
执行单元
EUN
222.N
执行单元
EUL
222.L
层级1
高速
缓存
223
非核心
资源
230
256
252
258
MISS
253
REPLAY
重排缓冲器 224
257
255
收回单元 225
WB

图2

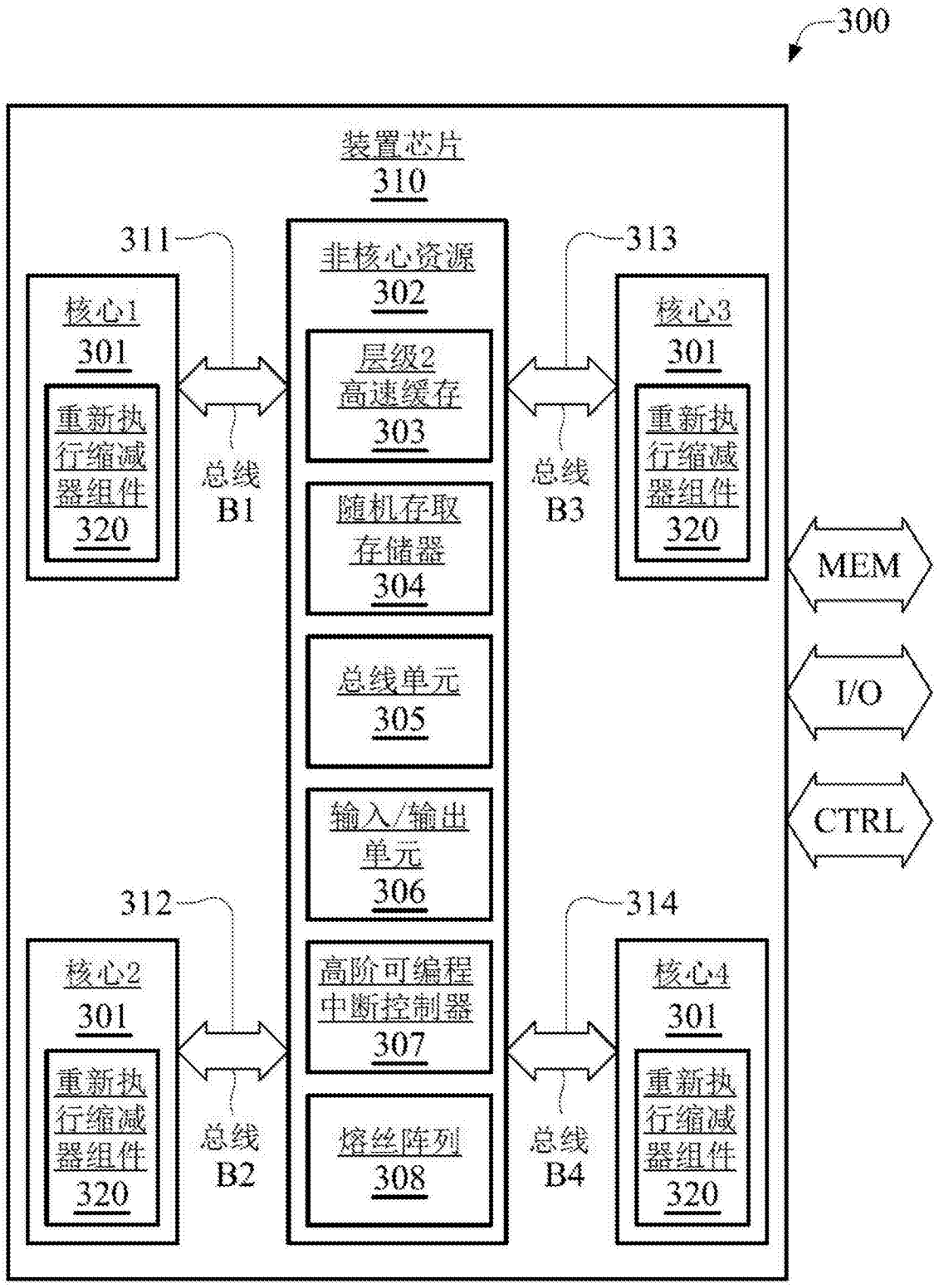

300
装置芯片
310
311
核心1
301
重新执行缩减器组件
320
总线
B1
非核心资源
302
层级2高速缓存
303
随机存取存储器
304
总线单元
305
输入/输出单元
306
高阶可编程中断控制器
307
熔丝阵列
308
313
核心3
301
重新执行缩减器组件
320
总线
B3
MEM
I/O
CTRL
312
核心2
301
重新执行缩减器组件
320
总线
B2
314
核心4
301
重新执行缩减器组件
320
总线
B4

图3

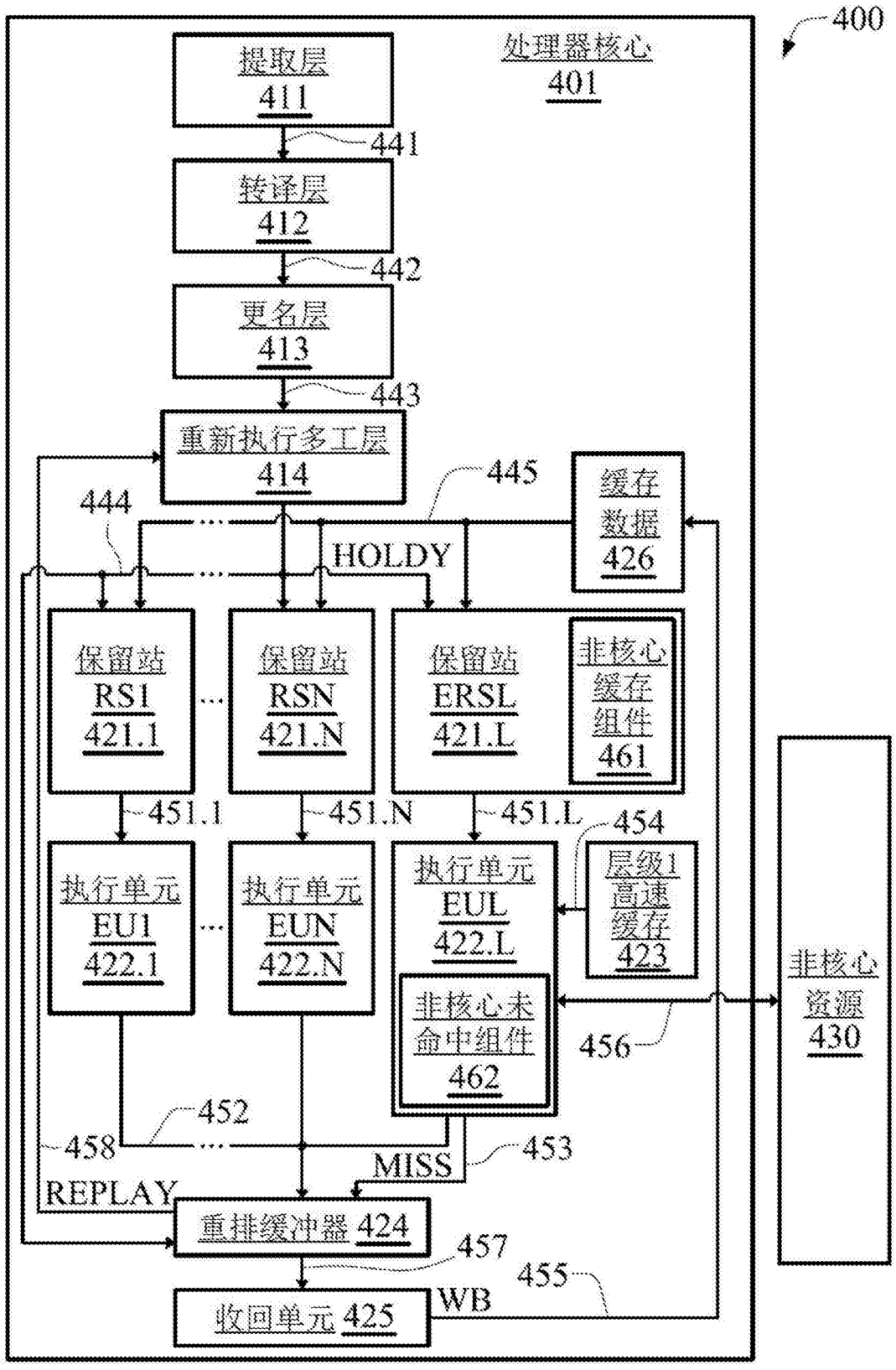

400
处理器核心
401
提取层
411
441
转译层
412
442
更名层
413
443
重新执行多工层
414
445
缓存
数据
426
444
HOLDY
保留站
RS1
421.1
保留站
RSN
421.N
保留站
ERSL
421.L
非核心
缓存
组件
461
451.1
451.N
451.L
454
执行单元
EU1
422.1
执行单元
EUN
422.N
执行单元
EUL
422.L
层级1
高速
缓存
423
非核心未
命中组件
462
456
非核心
资源
430
452
458
MISS
453
REPLAY
重排缓冲器 424
457
455
收回单元 425
WB

图4

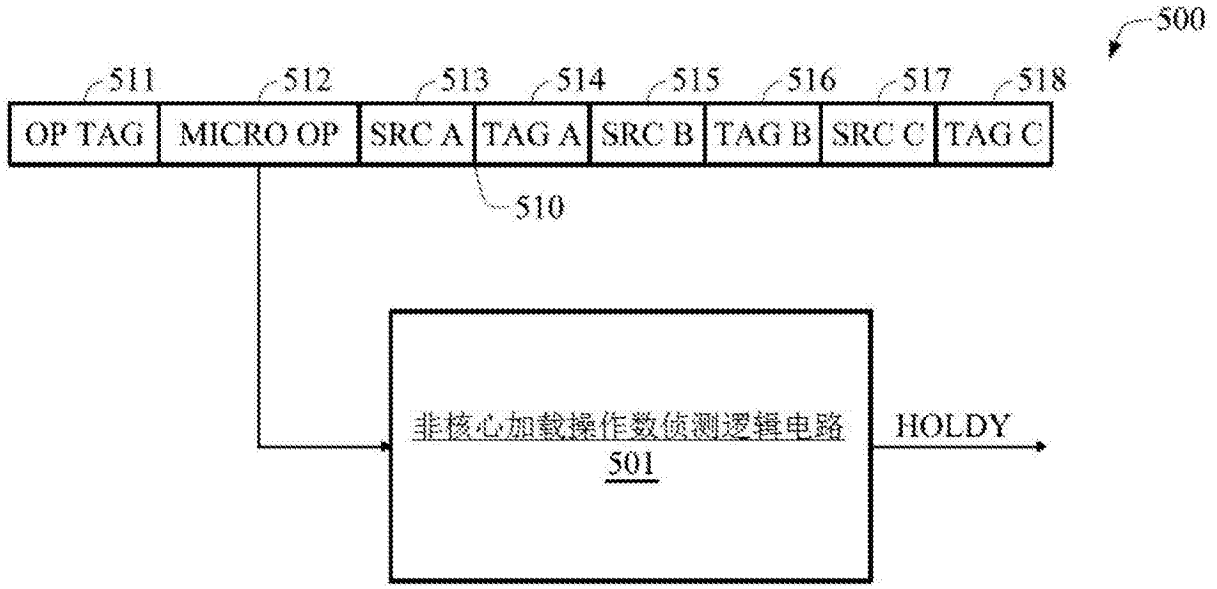

500
511
512
513
514
515
516
517
518
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
510
非核心加载操作数侦测逻辑电路
501
HOLDY

图5

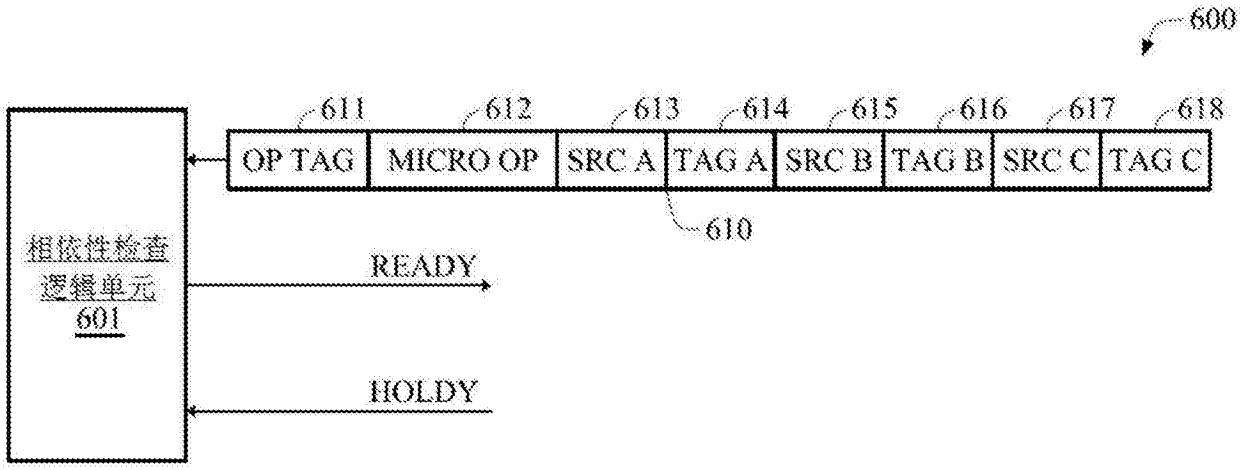

600
611
612
613
614
615
616
617
618
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
610
相依性检查
逻辑单元
601
READY
HOLDY

图6

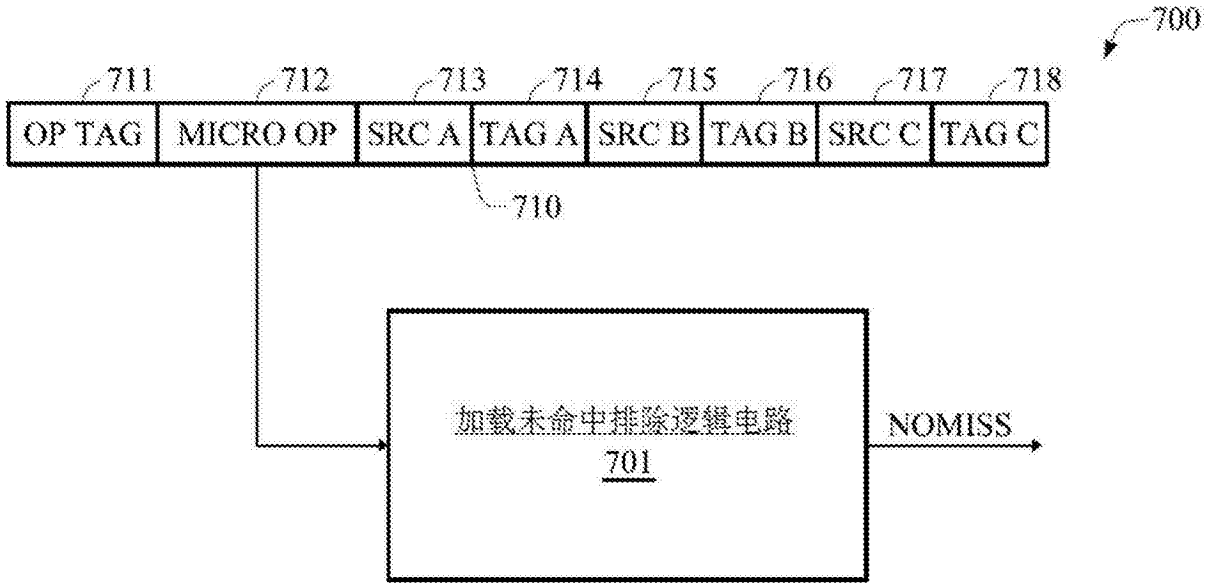

700
711
712
713
714
715
716
717
718
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
710
加载未命中排除逻辑电路
701
NOMISS

图7