[19] 中华人民共和国国家知识产权局

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

[12] 发明专利说明书

专利号 ZL 200480011945.8

[45] 授权公告日 2008 年 3 月 5 日

[11] 授权公告号 CN 100373330C

[22] 申请日 2004.1.9
[21] 申请号 200480011945.8
[30] 优先权
[32] 2003. 5. 2 [33] US [31] 10/429,159
[86] 国际申请 PCT/US2004/000483 2004.1.9
[87] 国际公布 WO2004/099978 英 2004.11.18
[85] 进入国家阶段日期 2005.11.2
[73] 专利权人 先进微装置公司
地址 美国加利福尼亚州
[72] 发明人 M·A·菲利普 J·K·皮克特
B·T·桑德尔
[56] 参考文献
US5966544A 1999.10.12
US5781752A 1998.7.14
WO0161480A1 2001.8.23
US5903741A 1999.5.11
WO0242902A2 2002.5.30
审查员 董 鑫
[74] 专利代理机构 北京纪凯知识产权代理有限公司
代理人 戈 泊 程 伟

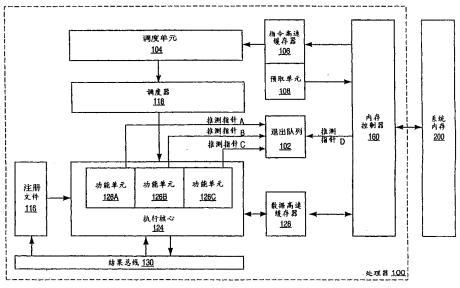
权利要求书 3 页 说明书 17 页 附图 4 页

[54] 发明名称

识别微处理器中的数据推测操作的推测指针

[57] 摘要

本发明涉及一种微处理器(100)，其可包括退出队列(102)和一个或多个数据推测验证单元。 配置每个数据推测验证单元以验证对操作执行的数据推测。 每个数据推测验证单元产生个别的识别未完成的操作的推测指针，这些未完成的操作已被数据推测验证单元验证。 配置退出队列(102)以根据从每个数据推测验证单元所接收的推测指针而选择性地退出操作。



1. 一种处理器（100），包括：

一个或多个数据推测验证单元，其被配置以验证对操作所执行的数据推测，并产生个别的推测指针以用于识别已由该一个或多个数据推测验证单元的其中一个验证的数据推测的未完成操作，各数据推测验证单元是包含在个别的功能单元（126）中；以及

退出队列单元，其被耦接至该一个或多个数据推测验证单元，并被配置以接收来自该一个或多个数据推测验证单元的每个单元的推测指针，及被配置以根据从该一个或多个数据推测验证单元的每个单元所接收的该推测指针而选择性地退出操作。

2. 如权利要求 1 所述的处理器（100），其中该功能单元（126）的其中一个包括加载存储单元（126C），其中，该加载存储单元（126C）包括该一个或多个数据推测验证单元中的一个，其中包含在该加载存储单元（126C）中的该数据推测验证单元被配置以验证相依性推测。

3. 如权利要求 1 所述的处理器（100），其中，该功能单元（126）中被配置以执行非内存操作的一个功能单元包括该一个或多个数据推测验证单元中的一个，其中包含在该功能单元（126）中的该数据推测验证单元被配置以验证推测非内存操作结果的数据推测。

4. 如权利要求 1 所述的处理器（100），其中该一个或多个数据推测验证单元中的一个单元被配置以接收识别多个未完成操作的每一个操作的信息，对该未完成操作已经执行由该一个或多个数据推测验证单元的其中一个所验证的一种类型的数据推测；以及

其中该一个或多个数据推测验证单元中的该一个单元被配置以识别哪些未完成的操作已经被验证，该验证是通过该一个或多个数据推测单元中的该一个单元通过提前个别推测指针以识别要比已经验证数据推测的最迟未完成操作更迟并且不比已经执行该类型的数据推测的最早未完成操作更迟的操作而完成的。

5.一种计算机系统（900），包括：

内存（200）；以及

耦接到该内存（200）的处理器（100）；

其特征在于：

该处理器（100）包括一个或多个数据推测验证单元，其被配置以验证对于操作所执行的数据推测，并且产生个别的推测指针用于识别已通过该一个或多个数据推测验证单元中的一个来验证数据推测的未完成操作，各数据推测验证单元是包含在个别的功能单元（126）中；以及

该处理器（100）进一步包括退出队列（102），其被耦接至该一个或多个数据推测验证单元，并被配置以接收来自该一个或多个数据推测验证单元的每个单元的推测指针，及被配置以根据从该一个或多个数据推测验证单元的每个单元所接收的该推测指针而选择地退出操作。

6.如权利要求5所述的计算机系统（900），其中该功能单元（126）的其中一个包括加载存储单元（126C），其中，该加载存储单元（126C）包括该一个或多个数据推测验证单元中的一个单元，其中包含在该加载存储单元（126C）中的该数据推测验证单元被配置以验证相依性推测。

7.如权利要求5所述的计算机系统（900），其中，该功能单元（126）中被配置以执行非内存操作的一个功能单元包括该一个或多个数据推测验证单元中的一个单元，其中包含在该功能单元（126）中的该数据推测验证单元被配置以验证推测非内存操作结果的数据推测。

8.一种用于选择性退出已执行数据推测的操作的方法，包括：

执行用于操作的数据推测；

验证单元验证所执行的用于该操作的该数据推测；

响应该验证，该验证单元产生推测指针，该推测指针指示对该验证单元而言该操作是非数据推测；以及

响应指示对该验证单元而言该操作是非数据推测的推测指针，而退出该操作。

9.如权利要求 8 所述的方法，进一步包括另一验证单元，其验证对另一操作执行的数据推测，并产生另一推测指针，其指示该另一操作对该另一验证单元而言是非数据推测的；

其中所述退出该操作是根据该推测指针和该另一推测指针。

10.如权利要求 8 所述的方法，其中所述产生指示对该验证单元而言该操作是非数据推测的该推测指针，是取决于该操作是已执行由该验证单元验证的任一类型的数据推测的最迟操作。

识别微处理器中的数据推测操作的推测指针

技术领域

本发明涉及微处理器领域，并且尤其涉及在微处理器中执行数据推测。

背景技术

超标量微处理器通过同时执行多个指令以及通过使用符合其设计的最短可行的时钟周期，而达到高性能。然而，各指令之间的数据和控制流相依性可以限制在任何给定的时间可发出多少个指令。因此，有些微处理器为了达成额外的增益，而支持推测执行功能。

其中一种类型的推测是控制流推测。控制流推测推测程序控制将进行的方向。例如，可以使用分支推测以推测是否将采用一个分支。可以使用许多种类型的分支推测，范围从每次仅作相同推测的方法，到那些在程序中保留先前各分支的复杂历史记录，而基于历史来作推测的方法。可通过硬件最佳化、编译器最佳化、或者两者兼而有之，而有助于分支推测。根据由分支推测机构所提供的推测，可以推测地取得指令并执行指令。当最后评估分支指令时，可验证该分支推测。若该推测不正确，则可撤销基于该不正确的推测而推测执行的任何指令。

另一种类型的推测是数据推测，该数据推测推测数据值。所提出的数据推测的类型包括以推测方式产生用于内存操作的地址，以及推测地产生用于计算机操作的数据值。与控制推测相同，最后评估用来推测产生值的基本（underlying）状况，允许可以验证或取消。

发明内容

本发明公开了根据与微处理器中各种验证单元有关的推测指针，而用于退出操作的各种方法和系统的实施例，该推测指针识别哪一个操作为数据推测操作。在一个实施例中，微处理器可包括退出队列和

一个或多个数据推测验证单元。配置各数据推测验证单元以验证对操作所执行的数据推测。每个数据推测验证单元产生用来识别未完成（outstanding）的操作的个别推测指针，对这些未完成的操作数据推测已由数据推测验证单元所验证。配置退出队列以根据从各数据推测验证单元所接收的推测指针而选择性地退出各操作。

在一个实施例中，其中数据推测验证单元可包含于加载存储单元中。可配置这样的数据推测验证单元以验证数据推测的类型，譬如相依性推测、地址推测、和/或数据推测。例如，可配置包含于加载存储单元中的数据推测验证单元，以验证推测较迟加载操作（younger load operation）并不根据（依赖于）具有未经计算地址的较早存储操作（older store operation）的相依性推测。同样地，可配置包含于加载存储单元中的数据推测验证单元，以验证推测较迟加载操作将等同于较早存储操作的来源的相依性推测。

另外一个数据推测验证单元可包含于功能单元中，该功能单元被配置以执行非内存操作。可配置这样的数据推测验证单元以验证推测非内存操作结果的数据推测。而另外一个数据推测验证单元可包含于内存控制器中，并被配置以验证内存推测。

在一个实施例中，可配置其中一个数据推测验证单元以接收识别每个未完成的操作的信息，对于这些未完成的操作已执行由该数据推测验证单元所验证的一种类型的数据推测。可配置该数据推测验证单元以验证哪一个未完成的操作已由该数据推测验证单元验证，通过前进（advance）个别的推测指针以识别一个操作，该操作要比已验证了一种数据推测类型的该最迟的未完成的操作更迟，并且要比已执行了该种数据推测类型的另一个未完成的操作更早的操作。若目前没有由该数据推测验证单元识别为数据推测的操作，则该数据推测验证单元可设定其个别推测指针的值以指示目前没有未完成的操作相对于该数据推测验证单元是数据推测。

在退出队列接收多个推测指针的实施例中，可配置退出队列以通过判断该操作是否要早于由所有推测指针识别是非数据推测的最迟操作，而判定操作是否退出。

在计算机系统中可包括微处理器，该微处理器包括一个或多个产

生推测指针的数据推测验证单元，以及根据这些推测指针而退出操作的退出队列。

在一些实施例中，一种方法可以包含：对于操作执行（perform）数据推测；验证单元验证对操作所执行的数据推测；验证单元产生推测指针以指示经验证后对验证单元来说不是数据推测的操作；以及针对指出那些对验证单元而言不是数据推测操作的推测指针，而退出该操作作为响应。

这样的方法也可包含一个或多个其它的验证单元，其验证对于其它操作所执行的数据推测，并产生其它的推测指针，其指示对验证单元而言其它的操作不是数据推测的其它操作。退出此操作也许要依赖于（根据）所有的推测指针。各验证单元可验证不同类型的数据推测。

附图说明

当结合附图考虑下面详细的描述时，可以得到本发明的更好的理解，其中：

图 1 表示依照本发明的微处理器；

图 2A 表示依照一个实施例，如何提前推测指针的流程图。

图 2B 表示依照一个实施例，根据现用推测指针条件退出操作的方法的流程图。

图 3 表示依照一个实施例例示的计算机系统。

图 4 表示依照另一个实施例的另一例示的计算机系统。

具体实施方式

虽然本发明易于作各种的修改和替代（可选）形式，参考附图例示的方式而详细说明了本发明的特定实施例。然而，应了解到此处特定实施例的说明并不是用来限制本发明为所公开的特定形式，而是相反地，本发明将包括所有落在所附权利要求所限定的本发明的精神和范围内的修改、等效和替代。应注意的是，标题的目的是仅用来组织本说明书，而并非想用来限制或解释说明的内容或权利要求。另外，要注意的是，单词“可以、也许（may）”是以容许的意义（即，具有潜在性（potential to），能够（being able to），而非以命令的意义

（即必须（must））而使用于整个申请。术语“包括（include）”及其衍生词意味着“包括，但不局限于”。术语“连接（connected）”意味着“直接或者不直接连接”，而术语“耦接（coupled）”意味着“直接或者不直接耦接”。

图 1 是微处理器 100 的一个实施例的方块图。配置微处理器 100 以执行存储于系统内存 200 中的指令。许多的这些指令操作存储于系统内存 200 中的数据。值得注意的是系统内存 200 可物理（physically）分布于整个计算机系统，并可由一个或多个微处理器 100 所存取。

微处理器 100 可包括指令高速缓存器 106 和数据高速缓存器 128。微处理器 100 可包括耦接到指令高速缓存器 106 的预取单元 108。可配置调度单元（dispatch unit）104 以接收从指令高速缓存器 106 来的指令，并调度操作给调度器（scheduler）118。一个或多个调度器 118 可耦接以接收从调度单元 104 来的分派操作，并发出操作给一个或多个执行核心 124。执行核心 124 可每一个都包括加载/存储单元配置以执行存取到数据高速缓存器 128。由执行核心 124 所产生的结果可输出到结果总线 130。可使用这些结果作为用于后续发出指令的操作域值和/或存储到注册（register）文件 116。退出队列 102 可耦接到调度器 118 和调度单元 104。可配置退出队列 102 以判定何时可退出每个发出的操作。在一个实施例中，可设计微处理器 100 与 x86 架构兼容。应注意的是微处理器 100 也可包括许多其它的组件。例如，微处理器 100 可包括分支推测单元（图中未表示）。

在指令由调度单元 104 接收前，指令高速缓存器 106 可暂时存储这些指令。可通过从系统内存 200 经过预取单元 108 预先取得指令码，而将指令码提供给指令高速缓存器 106。可以各种配置方式（例如，集合相关联、完全相关联、或直接映射），而实施指令高速缓存器 106。在一些实施例中，可以有多个层级（level）的指令和/或数据高速缓存器 106 和 128。可如图中所示，某些层级可与微处理器 100 整合（integrated），而高速缓存器的其它层级可设于微处理器的外部。

预取单元 108 可从系统内存 200 预先取得指令码而存储于指令高速缓存器 106 内。在一个实施例中，可配置预取单元 108 以将码从系统内存 200 突发（burst）到指令高速缓存器 106 中。预取单元 108 可

使用多种特定码的预先取得技术和算法。

调度单元 104 可输出信号，其包括可由执行核心 124 所执行的位编码操作以及操作数地址信息、直接数据（immediate data）、和/或位移数据（displacement data）。在一些实施例中，调度单元 104 可包括译码电路（图中未表示）用来将某些指令译码成可在执行核心 124 内执行的操作。简单的指令可对应于单一操作。在一些实施例中，更复杂的指令可对应于多个操作。若操作包含更新寄存器，则可保留注册文件 116 内的寄存器位置（例如，在译码该操作后）以存储推测寄存器状态（在一可选实施例中，可以使用重序缓冲器（reorder buffer）以存储对于每个寄存器的一个或多个推测寄存器状态）。寄存器映射可将来源和目的地操作数的逻辑寄存器名称转译为实际寄存器名称，以便促使寄存器重新命名。寄存器映射可追踪在注册文件 116 内的哪些个寄存器现正被分配。

图 1 的处理器 100 支持不按序的执行（out of order execution）。退出队列 102 可保持追踪用于寄存器读取和写入操作的原始的程序序列，允许推测指令执行和分支错误推测恢复，以及促进精确的排除（exception）。可以先进先出的配置方式执行退出队列 102，其中当它们有效时，操作移到缓冲器的“底部”，留出空间给新的在队列的“顶部”的入口。退出队列 102 可以退出操作，以响应于完成执行的操作及在任何操作上的任何数据或控制推测，直到包括依照程序命令的操作被检验。当退出了在实际寄存器中产生值的操作时，退出队列 102 可将该实际寄存器的推测状态指定为处理器 100 的架构状态。在一些实施例中，退出队列 102 可执行为重序缓冲器的一部分。这样的重序缓冲器也可提供用于推测寄存器状态的数据值存储，以便支持寄存器重新命名。值得注意的是，在其它的实施例中，退出队列 102 可不提供任何的数据值存储。代替地（反而），当退出操作时，退出队列 102 可解除分配在注册文件 116 中的寄存器，该注册文件 116 不再需要存储推测寄存器状态，并提供信号给寄存器映射以指示哪些寄存器现正闲置着。通过在注册文件 116 内维持推测的寄存器状态（或者，在可选实施例中，在重序缓冲器内），直到产生这些状态的每个操作有效为止，若分支推测为不正确，则在注册文件 116 中沿着错误推测路径而

推测执行操作的结果可以无效。

若所需的特定操作的操作数为寄存器位置，则可传递（route）寄存器位置信息给寄存器映射（或重序缓冲器）。例如，在 x86 架构中，有 8 个 32 位逻辑寄存器（例如，EAX、EBX、ECX、EDX、EBP、ESI、EDI 和 ESP）。物理注册文件 116（或重序缓冲器）包括存储改变这些逻辑暂存的内容的结果，允许不按序执行。可分配于注册文件 116 中的物理寄存器，以存储判定修正其中一个逻辑寄存器内容的每个操作的结果。因此，在执行特定程序期间的不同点，注册文件 116（或者，在可选实施例中的重序缓冲器）可具有一个或多个包含给定逻辑寄存器所推测执行内容的寄存器。

寄存器映射可将物理寄存器指定为指定的用于某一操作的目的地操作数的特定逻辑寄存器。调度单元 104 可判定注册文件 116 具有一个或多个先前分配指定给在给定操作中特定为来源操作数的逻辑寄存器的物理寄存器。寄存器映射可提供标记，用于标记最新指定给该逻辑寄存器的物理寄存器。使用该标记以存取于该注册文件 116 中的操作数的数据值，或通过传送在结果总线 130 上的结果而接收数据值。若该操作数对应于内存位置，则可通过加载/存储单元 222 在结果总线（用于结果传送和/或存储于注册文件 116 中）上提供操作数值。当由其中一个调度器 118 发出操作时，可提供操作数值给一个或多个执行核心 124。值得注意的是在可选的实施例中，当调度操作时可提供操作数值给对应的调度器 118（代替了当发出操作时提供给对应的执行核心 124）。

提供于调度单元 104 的输出的位编码操作和立即数据可路由（传递）给一个或多个调度器 118。应注意的是在此处所使用的，调度器是一种装置，其检测何时操作准备用于执行，以及发出准备好的操作给一个或多个功能单元。例如，保留站（reservation station）为一个调度器。在一个调度器或一组调度器中的操作也可称为指令中或操作窗口中或调度（scheduling）窗口中的操作。每个调度器 118 能够保持几个等待发送到执行核心 124 的待决（pending）操作的操作信息（例如，位编码执行位以及操作数值、操作数标记、和/或实时数据）。在一些实施例中，各调度器 118 可不提供操作数值存储。代替地，每个

调度器可监视发出的操作和在注册文件 116 中可用的结果，以便判定何时将由功能单元 126（从注册文件 116 或结果总线 130）读取操作数值。在一些实施例中，各调度器 118 可与专用的功能单元 126 相关。在其它的实施例中，单一调度器 118 可发出操作给不止--个的功能单元 126。

可提供调度器 118 以暂时存储将由执行核心 124 所执行的操作信息。如前所述，各调度器 118 可存储即将执行操作的操作信息。此外，各调度器可存储已经执行但也许仍再发出的操作的操作信息。可取得任何执行所需的操作数时，可发送操作给执行核心 124 以执行。因此，执行操作的次序可不相同于原来程序指令序列的次序。包含数据推测的操作可维持在调度器 118 中，至少直到这些操作变成非推测为止，以便若该数据推测为不正确的话，可再发出这些操作。

在一个实施例中，各执行核心 124 可包括多个功能单元 126（例如，如图 1 中所示的功能单元 126A 到 126C）。可配置诸如 126A 的某些功能单元以执行加法和减法的整数算术操作，以及位移、旋转、逻辑操作以及分支操作。可配置诸如 126B 的其它的功能单元以适合浮点操作。可配置诸如 126A 的一个或多个功能单元以执行由诸如 126C 的功能单元所执行的加载和存储内存操作的地址的产生，该诸如 126C 的功能单元执行加载和存储操作以存取存储在数据高速缓存器 128 中和/或系统内存中的数据。在一个实施例中，这样的加载存储单元 126C 可配置有加载/存储缓冲器，该加载/存储缓冲器具有多个用于即将执行加载和/或存储的数据和地址信息的存储位置。

一个或多个功能单元 126 也可提供与条件分支推测单元执行的相关的分支指令的信息，而使得若分支为错误推测，则分支推测单元可去除（flush）接续于（subsequent to）错误推测分支的已进入指令处理管线的指令，以及对预取单元 108 重新定向。然后重新定向的预取单元 108 可开始从指令高速缓存器 106 或系统内存 200 取得正确的指令组。在这样的情况下，可丢弃错误推测分支指令后所发生的原先程序序列中的指令结果，包括以推测方式执行和暂时存储于注册文件 116 中的指令结果。

若更新了寄存器值的话，则可以由输出执行核心 124 内的功能单

元 126 所产生的结果在结果总线 130 上给注册文件 116。若改变了内存位置的内容，则在执行核心 124 内所产生的结果可提供给加载/存储单元 126C。

数据高速缓存器 128 为高速缓存器，提供用来暂时存储转移在执行核心 124 和系统内存 200 之间的数据。如同上述的指令高速缓存器 106，指令高速缓存器 128 也可以不同的特定内存配置实施，包括一组集合相关的配置。此外，在一些实施例中，可实现指令高速缓存器 106 和指令高速缓存器 128 结合在一体的高速缓存器中。

在一些实施例中，处理器 100 可包括整合的内存控制器 160，允许微处理器直接与系统内存 200 接口。在其它的实施例中，内存控制器 160 可包含在把处理器 100 间接耦接到系统内存 200 的总线桥中。

数据推测

如其中所述，若可以发现数据值是不正确而结果要重新计算的话，则数据值是推测性的。推测的数据值是无法确实识别正确与否的数据值。若数据值为已执行的某一数据推测操作的结果，或若该数据值依赖于另一个推测数据值的话（例如，若数据值是由具有一个或多个推测的操作数的操作所产生的结果），则可能要再计算数据值。

在处理器 100 内的各种机构可执行数据推测。例如，调度单元 104、内存控制器 160、和/或一个或多个功能单元 126 可各自执行用于特定操作的数据推测。调度单元 104 可检测一个操作的结果可作为用于另一个操作的推测的操作数的情况。例如，调度单元可推测加载操作将通过先前的存储操作，而存取存储于数据高速缓存器 128 的数据。调度单元 104 可以反应地识别存储于用来作为存储操作的来源的寄存器中的数据值，可作为加载操作的推测结果。此类型的数据推测此处称为相依性推测。可通过连结（link）存储操作的来源作为操作推测的操作数来源，该操作指定该加载操作结果作为操作数，而将相依性推测扩展在调度单元 104 中。可通过允许某些加载跳过具有未计算地址的存储，即，通过推测较迟加载并非依赖于较早的存储，而可在加载存储单元 126C 中执行另一种类型的相依性推测。

在多重处理的系统中，内存控制器 160 可执行一致性检核

（coherency check）以维持高速缓存器一致性。内存控制器 160 在完成与其它的微处理器高速缓存器一致性检查之前，内存控制器 160 可推测地从系统内存 200 返回复制的高速缓存器线。若一致性检核接着判定应取得的正确高速缓存器线的复制是正存储在另一个处理器的高速缓存器中，则从系统内存 200 推测性取得的高速缓存器线的复制可以无效。因此，从存取该高速缓存器线所产生的任何加载操作结果将是推测的，直到该一致性检核完成为止。其中这样的类型的推测称为内存推测。

调度单元 104 可通过推测操作的结果，而执行数据推测。例如，有些操作也许倾向于产生相同的结果，而因此每次处理这些操作中的一个操作时，该结果可以在由功能单元 126 实际执行操作之前，通过调度单元 104 而推测地产生。其中，这样类型的数据推测称为数据推测。值得注意的是数据推测也可执行于微处理器的其它部分（例如，在加载存储单元 126C 中）。

加载存储单元 126C 可根据推测地址推测地产生地址，以及基于较早处理的加载的模式尚未计算出地址的加载指令的结果。例如，若前面的 N 个加载操作目标定为地址 A1 到 AN，这些地址彼此以固定偏移 C 间隔开（例如，A1；A2=A1+C；…；AN=A（N-1）+C），则加载存储单元 126C 可推测地返回最近存取地址 AN 加上固定偏移 C 中的数据作为加载操作的结果。这种类型的数据推测此处称为地址推测。值得注意的是其它形式的地址推测可使用于其它的实施例中。

根据已经执行数据推测的操作结果的操作也可产生推测结果。例如，若使用地址推测以产生加载操作的推测结果，则任何执行使用加载的推测结果作为操作数的依附操作可产生推测结果，这些推测结果可通过其它的相关操作（dependent operation）而用作操作数。因此，若在加载操作中的基本推测判定为不正确，则相关操作的结果也许也是不正确，而因此也许将须再执行根据该加载的操作的整个相依性链，以便产生正确的结果。另一方面，若发现基本推测是正确的，则依附操作的结果也许是正确的（假定这些结果并不根据任何其它的推测值）。

当由功能单元执行多个操作时，可验证已执行数据推测的这些操

作。例如，可通过功能单元 126 来验证用来推测地产生操作的结果的数据推测，该功能单元 126 通过比较操作的实际结果与推测结果，而执行该操作。若数据推测为不正确，则因为已可取得正确的结果，可不必再执行这些操作。可不需要完全地执行而验证其它的操作。例如，若具有未经计算地址的加载从较早的存储传送（forwarding）其结果（例如，由于相依性或地址推测），则当计算加载地址时，可验证加载的推测结果。若数据推测是不正确的，则可必须再执行这样的操作（至少部分地执行）以便产生正确的结果。

因为可以需要再执行对于已执行数据推测的操作和它们的相关操作，则可配置退出队列 102 以仅退出已解决了任何基本数据推测的操作。如图 1 中所示，可配置用来验证数据推测的各机构（在此实施例中，内存控制器 160 和功能单元 126），以提供具有推测指针指示已验证数据推测的各操作的退出队列 102。每个推测指针可用特定的验证装置（means），通过使推测指针具有相等于由该验证装置所验证的最迟操作的标记值，识别已验证数据推测的操作。退出队列 102 可使用该推测指针以识别可退出哪些操作。配置以验证一种或多种类型数据推测的处理器 100 内的各种组件，此处称为数据推测验证单元。

每个推测指针可识别在基本指令流中的哪一点相对于微处理器的特定部分是非推测的。例如，由内存控制器 160 所产生的推测指针 D，可识别与内存控制器 160 相关的最迟非推测的操作，内存控制器 160 验证内存推测值。在一个实施例中，内存控制器 160 可产生推测指针 D，指向该已执行内存推测的最近验证的操作。在另一个实施例中，内存控制器 160 可产生推测指针 D，指向已执行仅在内存推测的最早非验证的操作前一个的操作。一般而言，推测指针 D 指示该退出队列 102，对内存控制器 160 而言有哪些操作不是推测的。

在一个实施例中功能单元 126A、126B 可个别执行整数和浮点操作。功能单元 126A 和 126B 可各自验证数据推测。在图 1 所示的实施例中，推测指针 A 和 B 分别识别哪些操作已经通过功能单元 126A 和 126B 验证。由执行加载和存储操作的功能单元 126（例如，加载存储单元 126C）而产生推测指针 C。推测指针 C 可识别哪些操作已经由加载存储单元 126C 所验证。可配置加载存储单元 126C 以验证地址、数据、和/或相依性推测。

如上所述，各推测指针的值取决于哪些操作已通过个别的数据推测验证单元而被验证。在一些实施例中，由处理器 100 的验证部分所产生的推测指针的值也可取决于识别这些已执行数据推测的操作的信息。例如，在一个实施例中，内存控制器 160 可追踪已执行内存推测的每个操作。当内存控制器 160 验证每个操作时，内存控制器 160 可提前推测指针 D 以识别达到已执行内存推测的下一个最迟操作的所有的操作相对于内存控制器 160 是非数据推测。在一个实施例中，内存控制器 160 可通过提前推测指针 D 指向由内存控制器 160 所追踪的下一个最迟推测操作，而识别这些操作。同样地，执行其它类型的数据推测的机构（例如，调度单元 104 和/或执行加载和存储操作的功能单元 126）也可追踪已执行数据推测的操作。在一些实施例中，至少一些这样的数据推测机构可提供此信息给被配置以验证数据推测的类型的数据推测验证单元（例如，加载存储单元 126C 和/或一个或多个的其它功能单元 126）。若现在没有数据推测操作由特定的其中一个数据推测验证单元所验证，则通过该验证单元所产生的推测指针可设定为一个值，该值指示所有的未完成的操作相对于该特定验证单元是非数据推测。

退出队列 102 可通过比较由推测指针所识别的操作流的部分而识别哪些操作可退出。由所有推测指针所识别为非推测的最早操作可以是可通过退出队列 102 所退出的最早操作。例如，假定操作 0 到 10（以操作 0 是程序顺序中最早操作，以及 10 是程序顺序中最晚操作，其中程序顺序是在任何的操作重新排序或不按序处理在处理器 100 内已经执行之前在程序中执行指令顺序）已经由调度单元 104 所调度。若推测指针 A 指示达到操作 6 的各操作为非数据推测，推测指针 B 指示达到操作 5 的各操作为非数据推测，推测指针 C 指示达到操作 3 的各操作为非数据推测，和推测指针 D 指示所有的未完成的操作为非数据推测（例如，因为没有对任何操作 0 到 10 执行内存推测），则退出队列 102 可识别达到操作 3 的这些操作作为该组可退出的操作组。值得注意的是操作退出也可根据一般退出限制，例如是否这些操作已经由功能单元 126 所执行，以及是否任何影响这些操作的控制推测已经成功地解决。例如，若数据推测操作判定为不正确并且需要再执行，而

该操作可通过一个或多个推测指针识别为非推测的，但是将不会被退出，直到该操作已经再执行为止。在一些实施例中，关于哪些操作已经执行和尚未执行，或者若需要的话则再执行的信息，可通过调度器118而提供给退出队列102。

图2A例示用来产生推测指针的方法的一个实施例。可通过其中一种数据推测验证单元，譬如内存控制器160和功能单元126，而至少部分地执行这样的方法。在201，执行操作的数据推测。在一些实施例中可通过将验证数据推测的微处理器的相同部分，而执行数据推测201。在203中若已验证了数据推测，则可提前推测指针如在205所示，以识别在201已执行数据推测为非推测的操作。

在一个实施例中，在205提前推测指针可包含提前推测指针以识别最近已验证的操作，指示所有到达该操作的操作和包括该操作相关于特定用于验证数据推测的机构为非数据推测的。在其它的实施例中，在205提前推测指针可包含提前推测指针以识别就在以程序顺序将由该特定的验证机构所验证的下一个数据推测操作之前的操作。例如，加载存储单元126C可追踪该加载存储单元已经对于哪些操作执行数据推测。每当该加载存储单元126C验证那些数据推测操作中的一个操作时，该加载存储单元126C可提前其推测指针以指示所有到达由加载存储单元执行数据推测的下一个操作的操作，对加载存储单元而言为非数据推测。其它的实施例可以以其它的方式提前指针。例如，功能单元可验证由调度单元所执行的数据推测的类型。在一些实施例中，功能单元也许不知已由调度单元执行该类型的数据推测的全部未完成的操作组。代替地，功能单元可仅知现在在该功能单元内的哪些未完成的操作为数据推测的。因此，响应于用于特定操作的验证数据推测，功能单元可提前该推测指针以识别在该功能单元内正早于最早的数据推测操作（如果有的话）的该未完成的操作。若在该功能单元内现在没有任何未完成的操作为数据推测的，则功能单元可更新其推测指针的值，以指示相关于该功能单元没有未完成的操作现在正是数据推测的。

图2B例示退出未完成的操作的一个方法实施例的流程图。在211，接收一个或多个推测指针。若接收了多个推测指针，则各推测指针可

识别为非数据推测的未完成的操作的不同部分。若任何推测指针指示特定的操作仍为数据推测的，则可不退出该操作，如在 213 所示。然而，若无任何的推测指针识别该操作可能是数据推测的，假定符合用于该操作退出的所有的其它预定必须条件，则可退出该操作，如 213 到 215 所示。

例示的计算机系统

图 3 表示计算机系统 900 的一个实施例的方块图，该计算机系统 900 包括处理器 100，其通过总线桥 902 耦接到各种系统组件。处理器 100 可包括一个或多个配置以产生推测指针的数据验证单元，以及配置以退出操作的退出队列，该操作是由推测指针识别为非数据推测的，如上所述。计算机系统的其它实施例是可行的并且是可以考虑的。在所描述的系统中，主内存 200 通过内存总线 906 耦接到总线桥 902，以及图形控制器 908 通过 AGP 总线 910 耦接到总线桥 902。多个 PCI 装置 912A 到 912B 通过 PCI 总线 914 耦接到总线桥 902。也可设有第二总线桥 916，通过 EISA/ISA 总线 920 提供电子接口给一个或多个 EISA 或 ISA 装置 918。在本实施例中，处理器 100 通过 CPU 总线 924 耦接到总线桥 902，以及可选的 L2 高速缓存器 928。在一些实施例中，处理器 100 可包括整合的 L1 高速缓存器（图中未表示）。

总线桥 902 提供处理器 100、主内存 200、图形控制器 908、以及连接到 PCI 总线 914 装置之间的界面。当操作由连接到总线桥 902 的其中一个装置所接收时，总线桥 902 识别该操作的目标（例如，特定的装置，或在 PCI 总线 914 的情况，该目标是在 PCI 总线 914 上）。总线桥 902 将该操作传递给目标装置。总线桥 902 通常从由来源装置或总线所使用的协议转译操作成由目标装置或总线所使用的协议。

除了为 PCI 总线 914 提供接口（interface）给 EISA/ISA 总线以外，第二总线桥 916 也可结合额外的功能性。可从外部接入，或者与第二总线桥 916 整合在一起的输入/输出控制器（图中未表示），也可包括在计算机系统 900 内，以提供对于键盘和鼠标 922，以及用于各种的串行端口和并行端口的操作支持。在其它的实施例中，外部高速缓存器单元（图中未表示）也可耦接到位于处理器 100 与总线桥 902 之

间的 CPU 总线 924。或者，外部高速缓存器可耦接到总线桥 902，以及用于外部高速缓存器的高速缓存器控制逻辑也可整合到总线桥 902。L2 高速缓存器 928 表示为处理器 100 的背侧（backside）配置。值得注意的是，L2 高速缓存器 928 可从处理器 100 分离，整合到具有处理器 100 的卡匣内（例如，槽 1 或槽 A），或甚至整合到具有处理器 100 的半导体基板上。

主内存 200 是存储应用程序的内存，并且处理器 100 主要从主内存 200 而执行。适当的主内存 200 可包括动态随机存取内存（DRAM）。例如，多个库的同步 DRAM（SDRAM）或总线 DRAM（RDRAM）可以是合适的。

PCI 装置 912A 到 912B 表示各种的外围装置，譬如网络接口卡、视频加速器、声频卡、硬盘机或软盘机或其控制器、小计算机系统接口（SCSI）适配器和电话卡。同样地，ISA 装置 918 例示了各种类型的外围装置，譬如调制解调器、声卡、和各种数据采集卡，譬如 GPIB 或区域总线接口卡。

设有图形控制器 908 以控制显示器 926 上的文件和图像的着色。图形控制器 908 可以通常在本领域已知的典型的图形加速器进行具体实施，对能够有效移位入主内存 200 或从主内存 200 移位出的三维数据结构进行着色。图形控制器 908 因此可以是 AGP 总线 910 的主控电路。图形控制器 908 能在该 AGP 总线 910 请求和接收到总线桥 902 内的目标接口的存取，由此而获得到主内存 200 的存取。专用图形（graphics）总线能适应从主内存 200 快速恢复的数据。对于某些操作，可进一步配置图形控制器 908 以在 AGP 总线 910 上产生 PCI 协议处理（transaction）。总线桥 902 的 AGP 接口可因此包括功能性以支持 AGP 协议处理以及 PCI 协议目标和起始器处理。显示器 926 为任何的电子显示器，影像或文件可呈现在此显示器上。适当的显示器 926 包括阴极射线管（CRT）、液晶显示器（LCD）等等。

应注意的是，在上述中虽然已使用了 AGP、PCI、和 ISA 或 EISA 总线作为例子，但是当需要时用任何的总线架构进行替代。更值得注意的是，计算机系统 900 可以是包括额外处理器（例如，处理器 100a 表示为计算机系统 900 的可选的组件）的多处理计算机系统。处理器

100a 可相似于处理器 100。尤其地，处理器 100a 可以是与处理器 100 完全相同的副本。处理器 100a 可通过独立的总线（图 3 中所示）而连接到总线桥 902，或可与处理器 100 共享 CPU 总线 924。而且，处理器 100a 可耦接到选用的相似于 L2 高速缓存器 928 的 L2 高速缓存器 928a。

现参照图 4，表示了计算机系统 900 的另一个实施例，可包括处理器 100，该处理器 100 具有一个或多个数据推测验证单元配置以产生推测指针和退出队列，该退出队列依照如上述的该推测指针而退出操作。其它的实施例也是可行的并且可以考虑的。在图 4 的实施例中，计算机系统 900 包括多个处理节点 1012A、1012B、1012C 和 1012D。每个处理节点通过包含在每个个别处理节点 1012A 到 1012D 内的内存控制器 1016A 到 1016D 而耦接到个别的内存 200A 到 200D。此外，处理节点 1012A 到 1012D 包括用来在处理节点 1012A 到 1012D 之间通讯的接口逻辑。例如，处理节点 1012A 包括用来与处理节点 1012B 通讯的接口逻辑 1018A、用来与处理节点 1012C 通讯的接口逻辑 1018B、以及用来与又另一处理节点（图中未表示）通讯的第三接口逻辑 1018C。同样地，处理节点 1012B 包括接口逻辑 1018D、1018E 和 1018F；处理节点 1012C 包括接口逻辑 1018G、1018H 和 1018I；以及处理节点 1012D 包括接口逻辑 1018J、1018K 和 1018L。处理节点 1012D 通过接口逻辑 1018 L 被耦接以与多个输入/输出装置（例如，在菊式链（daisy-chain）配置中的装置 1020A 到 1020B）通讯。其它的处理节点可与其它的 I/O 装置以相似的方式通讯。

处理节点 1012A 到 1012D 实施以封包为基础的连结以用于处理节点间通讯。在本实施例中，该连结是作为单方向线组来实施的（例如，线 1024A 用来从处理节点 1012A 到 1012B 传输封包和线 1024B 用来从处理节点 1012B 到 1012A 传输封包）。其它组的线 1024C 到 1024H 用来在其它处理节点之间传输封包，如图 4 中所示。一般而言，各组的线 1024 可包括一条或多条数据线；一条或多条对应于各数据线的时钟线，以及一条或多条指示传输的封包类型的控制线。对于处理节点间的通讯，该连结可以以高速缓存器一致性方式操作，或以用于在操作处理节点和 I/O 装置（或是譬如 PCI 总线或 ISA 总线的所知结构的 I/O 总线到总线桥）之间通讯的一致性方式来操作。另外，可使用如所示的

在 I/O 装置之间的菊式链结构而以非一致性方式操作连结。应注意的是将从一个处理节点传输到另一个处理节点的封包可经过一个或多个中间节点。例如，由处理节点 1012A 传输到处理节点 1012D 的封包可经过处理节点 1012B 或处理节点 1012C，如图 4 中所示。可使用任何适当的路线算法。计算机系统 900 的其它实施例可包括较图 4 中所示者更多或较少的处理节点。

一般而言，封包可以作为一个或多个位时间（bit time）而传输在各节点之间的线路 1024 上。该位时间可以是对应的时钟线上的时钟信号的上升沿或下降沿。封包可包括用于起使处理的命令封包、维持高速缓存器一致性探测封包、和从响应于探测和命令封包而来的反应封包。

除了内存控制器和接口逻辑外，处理节点 1012A 到 1012D 可包括一个或多个处理器。概括地说，处理节点包括至少一个处理器以及需要的可选择性地包括用于与内存和其它逻辑通讯的内存控制器。尤其地，每个处理节点 1012A 到 1012D 可包括各处理器 100 的一个或多个副本。外部接口单元可包括在节点内的接口逻辑 1018 以及内存控制器 1016。

内存 200A 到 200D 可包括任何适当的内存装置。例如，内存 200A 到 200D 可包括一个或多个总线（RAMBUS）DRAM（RDRAM）、同步 DRAM（SDRAM）、静态 RAM 等等。计算机系统 900 的地址空间划分在内存 200A 到 200D 之间。各处理节点 1012A 到 1012D 可包括用来判定哪个地址映射到内存 200A 到 200D 的哪个内存的内存映射，而因此决定具有特定地址的内存请求将传递到哪一个处理节点 1012A 到 1012D。在一个实施例中，对于计算机系统 900 内地址的一致点为耦接到内存存储有对应于该地址的字节的内存控制器 1016A 到 1016D。换句话说，内存控制器 1016A 到 1016D 负责确定各内存存取到对应的内存 200A 到 200D 以高速缓存器一致方式发生。内存控制器 1016A 到 1016D 可包括用来接口到内存 200A 到 200D 的控制电路。此外，内存控制器 1016A 到 1016D 可包括用来排列内存请求的请求队列。

接口逻辑 1018A 到 1018L 可包括各种缓冲器用于从连结接收封包并缓冲将传送到连结上的封包。计算机系统 900 可使用任何适当的用

于传输封胞的流程控制机构。例如，在一个实施例中，各接口逻辑 1018 存储位于连接接口的连结（link）的另一端的接收器内各种类型的缓冲器数量的数值。除非接收的接口逻辑具有闲置缓冲器以存储封包，否则接口逻辑不发送封包。当通过将封包向前传送（forward）而使接收缓冲器为闲置时，接收接口逻辑发出信号到发送接口逻辑以指示缓冲器已经闲置。这样的机构可称为“凭票（coupon-based）”系统。

I/O 装置 1020A 到 1020B 可以是任何适当的 I/O 装置。例如，I/O 装置 1020A 到 1020B 可以包括与其它计算机系统通讯的装置（例如，用网络接口卡或调制解调器），这些装置耦接到其它计算机系统。另外，I/O 装置 1020A 到 1020B 可包括视频加速器、声频卡、硬盘或软盘机或驱动控制器、小计算机系统接口（SCSI）适配器和电话卡、声卡、以及各种数据采集卡，譬如 GPIB 或区域总线接口卡。值得注意的是术语“I/O 装置”和术语“外围装置”在此处规定为同义词。

如其中所使用的术语“时钟周期”是指时间间隔，在此时间间隔中各级的指令处理路线完成其工作。通过内存组件（譬如寄存器或数组）依照定义时钟周期的时钟信号而获得指令和计算值。例如，内存组件可依照时钟信号的上升沿和下降沿而获得数值。

以上讨论说明信号为“确立的（asserted）”。信号可以定义为确立的当它输送代表一项特定信息的值。特定信号可定义成确立的当它输送二进制的 1 值，或者取而代的当它输送二进制的 0 值时。

一旦本领域技术人员完全了解该上述的公开内容，不同的变化及修改将变得显而易见。其意图在于下列的权利要求将被解释以包含所有这样的变化与修改。

工业应用性

本发明通常可应用于微处理器。

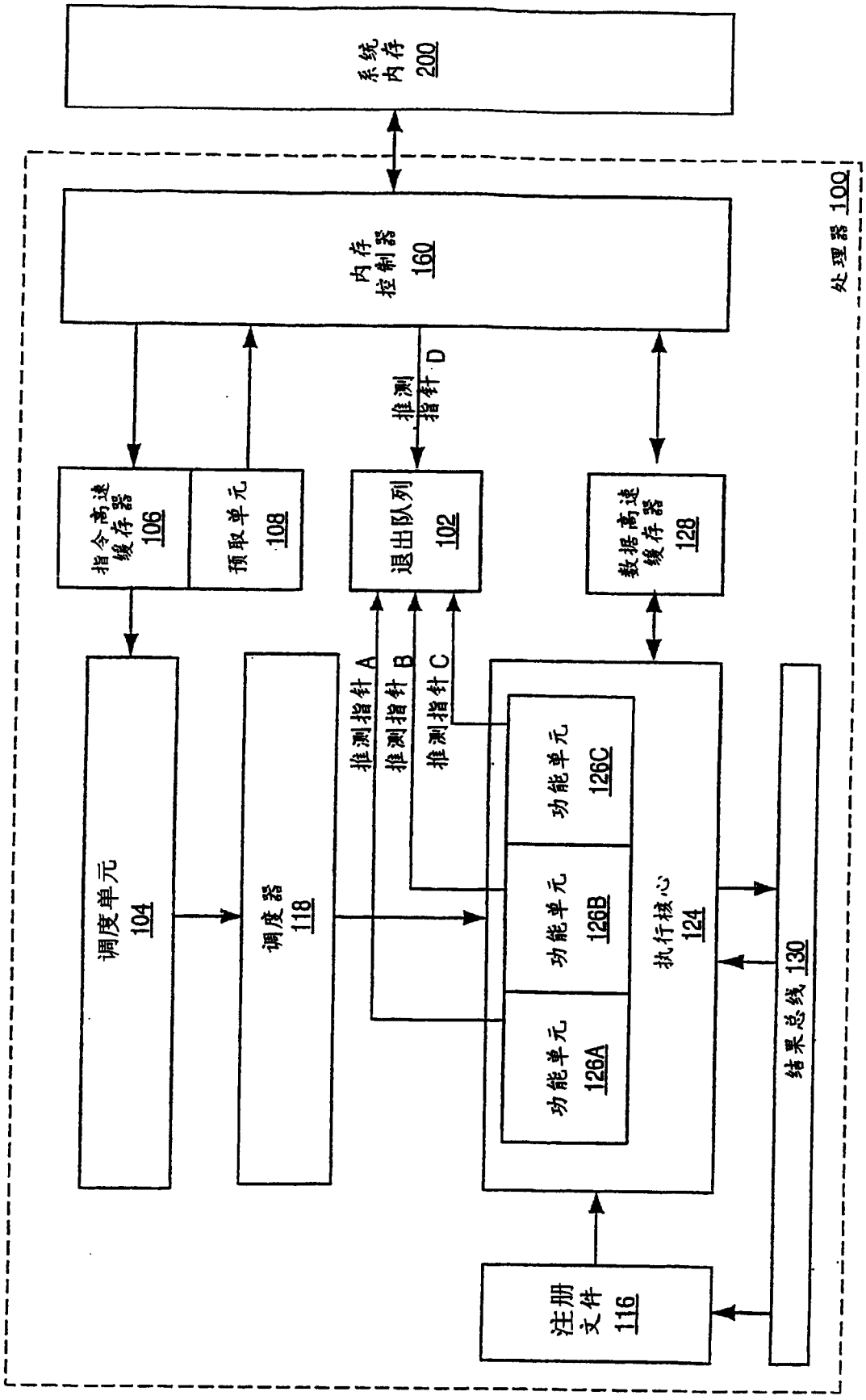

系统内存 200
处理器 100
内存控制器 160
推测指针 D
指令高速缓存器 106
预取单元 108
退出队列 102
数据高速缓存器 128
调度单元 104
调度器 118
推测指针 A
推测指针 B
推测指针 C
功能单元 126C
功能单元 126B
功能单元 126A
执行核心 124
结果总线 130
注册文件 116

图 1

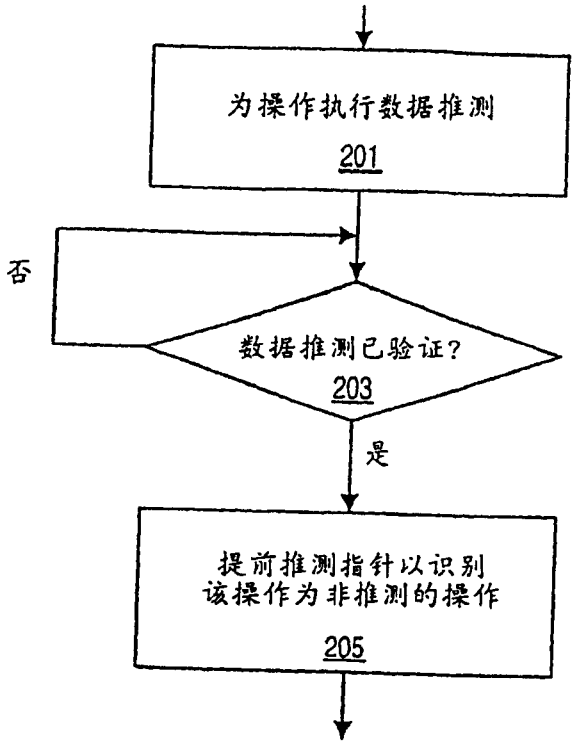

为操作执行数据推测
201
否
数据推测已验证?
203
是
提前推测指针以识别
该操作为非推测的操作
205

图2A

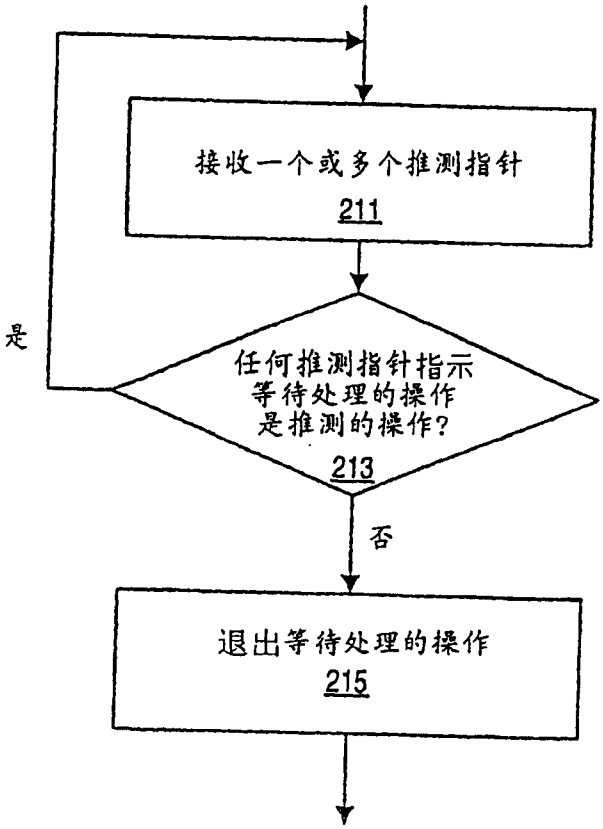

接收一个或多个推测指针
211
是
任何推测指针指示
等待处理的操作
是推测的操作?
213
否
退出等待处理的操作
215

图2B

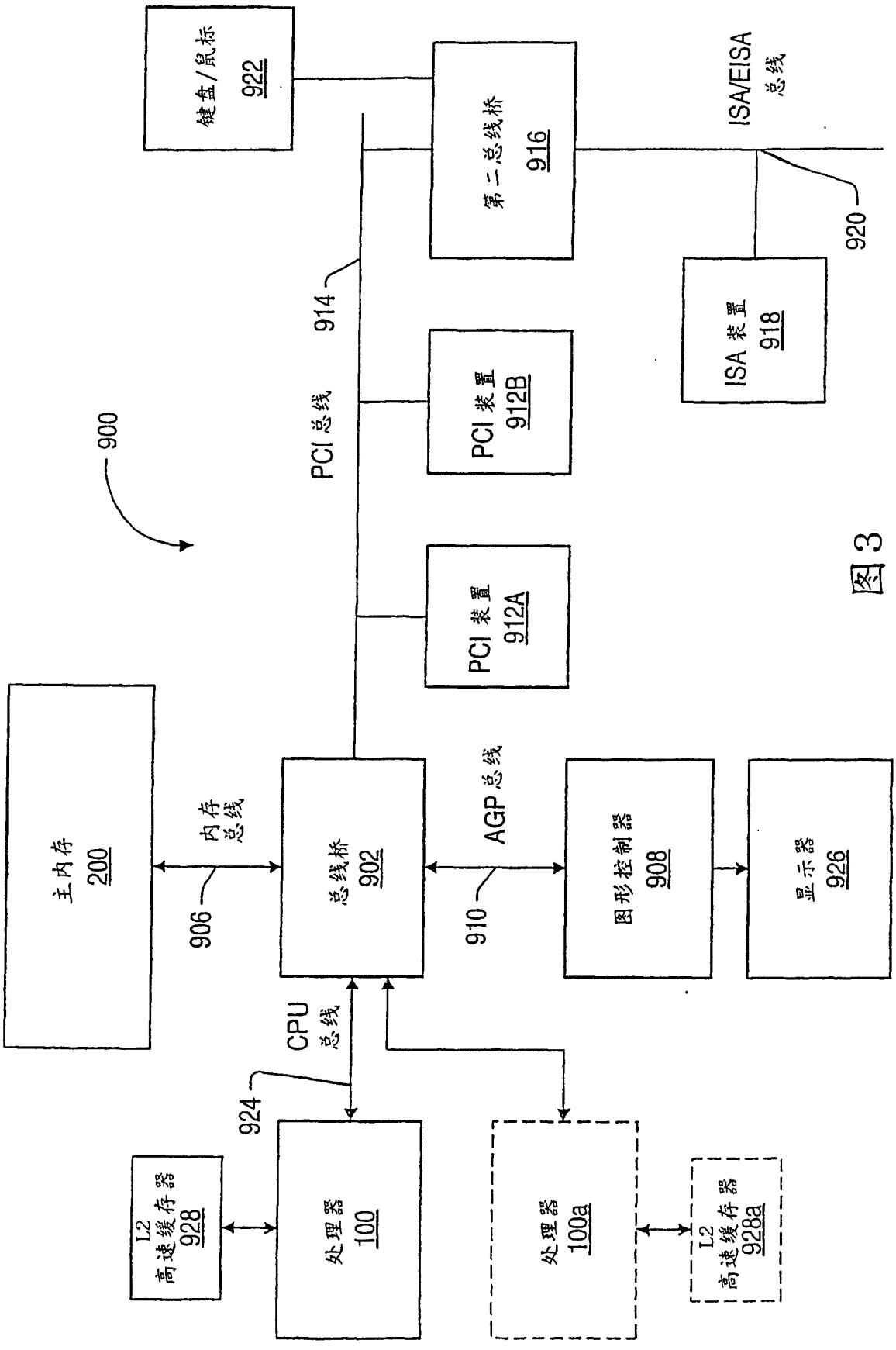

900
主内存
200
内存
总线
906
总线桥
902
AGP 总线
910
图形控制器
908
显示器
926
CPU
总线
924
处理器
100
L2
高速缓存器
928
处理器
100a
L2
高速缓存器
928a
PCI 总线
914
PCI 装置
912A
PCI 装置
912B
第二总线桥
916
ISA/EISA
总线
920
ISA 装置
918
键盘/鼠标
922

图 3

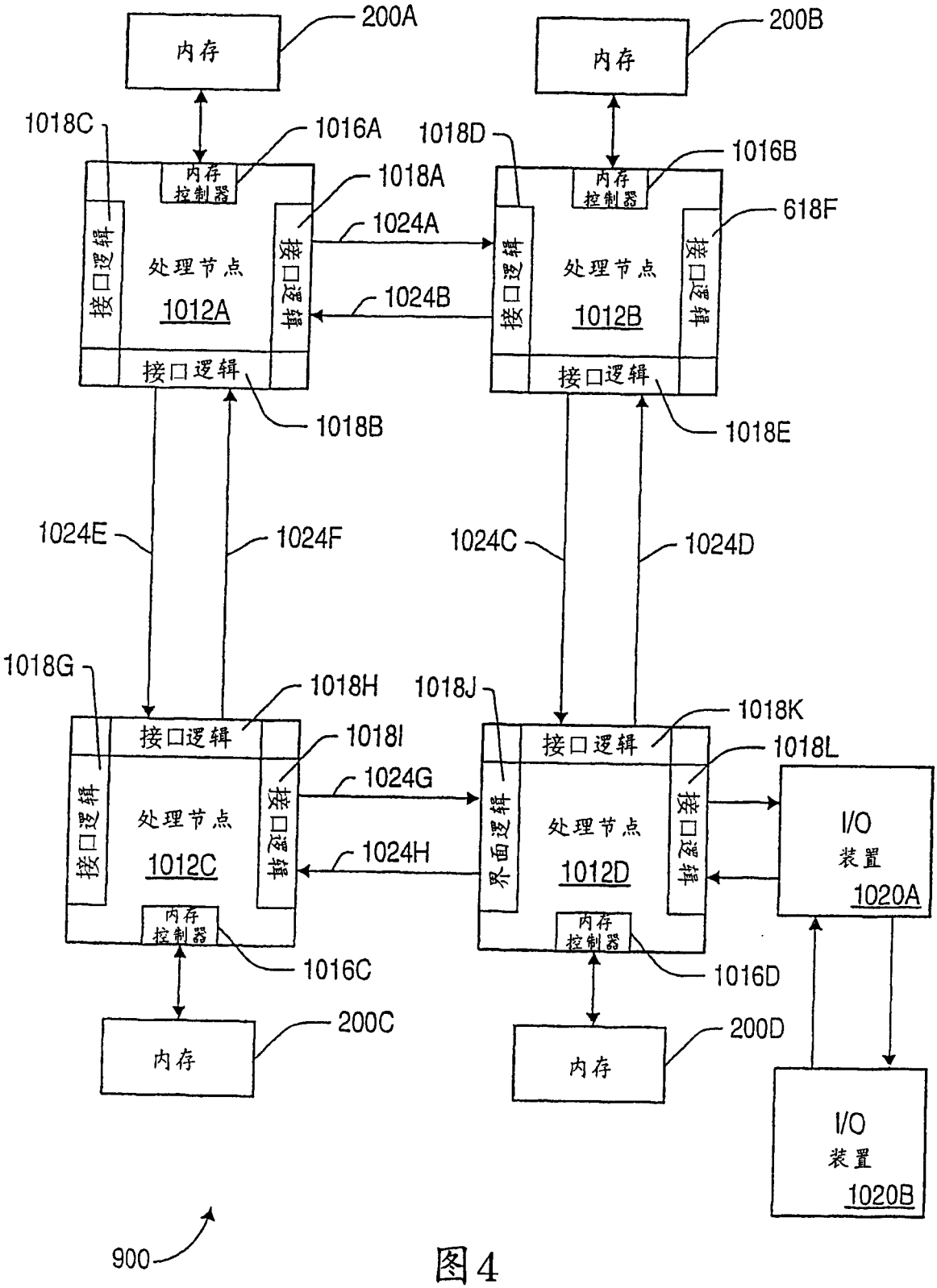

内存
200A
内存
200B
1018C
1016A
1018D
1016B
内存控制器
内存控制器
1018A
618F
1024A
接口逻辑
处理节点
1012A
接口逻辑
接口逻辑
处理节点
1012B
接口逻辑
1024B
接口逻辑
接口逻辑
1018B
1018E
1024E
1024F
1024C
1024D
1018G
1018H
1018J
1018K
接口逻辑
接口逻辑
1018I
1018L
1024G
接口逻辑
处理节点
1012C
界面逻辑
处理节点
1012D
接口逻辑
I/O
装置
1020A
1024H
内存控制器
内存控制器
1016C
1016D
200C
200D
内存
内存
I/O
装置
1020B
900

图4