

[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 200480028753.8

[45] 授权公告日 2008 年 7 月 30 日

[11] 授权公告号 CN 100407134C

[22] 申请日 2004.6.28

[21] 申请号 200480028753.8

[30] 优先权

[32] 2003.10.1 [33] US [31] 10/676,437

[86] 国际申请 PCT/US2004/020721 2004.6.28

[87] 国际公布 WO2005/041024 英 2005.5.6

[85] 进入国家阶段日期 2006.4.3

[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·阿尔叙普 G·W·什毛斯
J·K·皮克特 B·D·麦克茨茵
M·A·菲利波 B·T·桑德尔

[56] 参考文献

US6256727B1 2002.7.3

US6339822B1 2002.1.15

US6449714B1 2002.9.10

A 0.18 – MUM CMOS IA – 32 PROCESSOR
WITH A 4 – GHZ INTEGER EXECUTIVE UNIT.
HINTON G ET AL. IEEE JOURNAL OF SOLID-
STATE CIRCUITS, Vol. 36 No. 11. 2001
INSTRUCTION PATH COPROCESSORS.
CHOUYET AL. INTERNATIONAL SYMPOSIUM
ON COMPUTER ARCHITECTURE. 2000

审查员 程 琼

[74] 专利代理机构 北京纪凯知识产权代理有限公司

代理人 程 伟 王锦阳

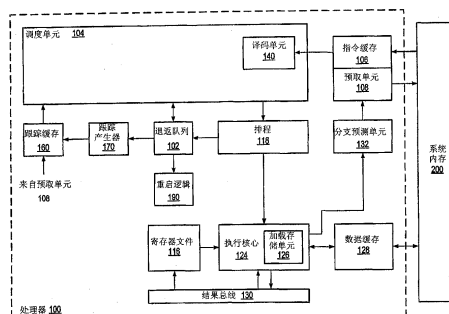
权利要求书 3 页 说明书 25 页 附图 5 页

[54] 发明名称

处置在基于跟踪缓存的处理器中的异常指令
的系统及方法

[57] 摘要

系统可以包含指令缓存(106)、包含多个跟踪缓存登录(162)的跟踪缓存(160)及连结至所述指令缓存(106)及所述跟踪缓存(160)的跟踪产生器(170)。所述跟踪产生器(170)可以经由配置成接收通过所述指令缓存(106)所输出的指令群组,而用于存储在其中一个所述多个跟踪缓存登录(162)内。所述跟踪产生器(170)可以经由配置以检测在所述指令群组内部的异常指令及经由配置以避免所述异常指令作为任何非异常指令而存储在相同的所述多个跟踪缓存登录(162)内的其中之一。



1. 一种处置在基于跟踪缓存的处理器中的异常指令的系统，其包括：

指令缓存（106）；

包含多个跟踪缓存登录（162）的跟踪缓存（160）；以及
连结至所述指令缓存（106）及所述跟踪缓存（160）的跟踪产生器（170）；

其中，所述跟踪产生器（170）配置成接收通过所述指令缓存（106）所输出的指令群组，以存储在所述多个跟踪缓存登录（162）的其中之一内，其中，所述跟踪产生器（170）配置成检测在所述指令群组内部的异常指令，及避免将所述异常指令存储在所述多个跟踪缓存登录（162）中的相同一个跟踪缓存登录内而成为任何非异常指令。

2. 如权利要求 1 所述的系统，其中，所述跟踪产生器（170）配置用于以至少部分译码的形式把指令存储在所述跟踪缓存（160）内。

3. 如权利要求 2 所述的系统，还包括连结至退返执行指令的退返队列（102），其中，所述退返队列（102）配置成避免从所述跟踪缓存（160）所取的任何指令的退返，直到包含在相同于所述指令的跟踪缓存登录（162）内的所有的指令准备好退返。

4. 如权利要求 2 所述的系统，还包括配置成发送从所述指令缓存（106）所接收的指令的调度单元（104），其中，所述调度单元（104）配置成检测在从所述指令缓存（106）所接收的指令群组内部的所述异常指令，及用以把所述异常指令的指示提供给所述跟踪产生器（170），其中，所述跟踪产生器（170）配置成检测所述异常指令，以响应于来自所述调度单元（104）的指示。

5. 如权利要求 2 所述的系统，其中，所述跟踪产生器（170）配置成把所述异常指令存储在不同于所述一个或多个其它指令的跟踪缓存

登录（162）内，其中，所述跟踪产生器（170）配置成不把具有所述异常指令的任何其它指令存储在所述不同的跟踪缓存登录（162）内。

6. 一种处置在基于跟踪缓存的处理器中的异常指令的方法，其包括：

跟踪产生器（170）接收用于存储在连结至所述跟踪产生器（170）的跟踪缓存（160）内部的跟踪缓存登录（162）内的指令群组；

所述跟踪产生器（170）将包含在所述指令群组中的一个或多个指令存储在所述跟踪缓存登录（162）内；

所述跟踪产生器（170）检测在所述指令群组内部的异常指令；以及

响应于所述检测，所述跟踪产生器（170）不将所述异常指令存储在具有所述一个或多个指令的所述跟踪缓存登录（162）内部。

7. 如权利要求 6 所述的方法，其中，所述跟踪产生器（170）以至少部分译码的形式把所述一个或多个指令存储在所述跟踪缓存登录（162）内。

8. 如权利要求 7 所述的方法，还包括退返队列（102）避免从所述跟踪缓存（160）所取的任何指令的退返，直到包含在与所述指令相同的跟踪缓存登录（162）内的所有指令准备好退返。

9. 如权利要求 8 所述的方法，还包括：

重启逻辑（190）监测自任何指令的退返以来所经过的周期数目；以及

若所述周期数目超过临界数目，则刷新从所述跟踪缓存（160）所取的处理管线执行的指令，及重新开始从所述指令缓存（106）的执行。

10. 一种计算机系统（400），其包括：

系统内存（404）；以及

连结至所述系统内存（404）的处理器（10）；

其特征在于所述处理器（10）包含：
指令缓存（106）；
包含多个跟踪缓存登录（162）的跟踪缓存（160）；以及
连结至所述指令缓存（106）及所述跟踪缓存（160）的跟踪产生器（170）；

其中，所述跟踪产生器（170）配置成接收由所述指令缓存（106）所输出的指令群组，以存储在所述多个跟踪缓存登录（162）的其中之一内，其中，所述跟踪产生器（170）配置成检测在所述指令群组内部的异常指令，及避免将所述异常指令存储在所述多个跟踪缓存登录（162）中的相同一个跟踪缓存登录内而成为任何非异常指令。

处置在基于跟踪缓存的处理器中的异常指令的系统及方法

技术领域

本发明系关于处理器的领域，尤系关于具有跟踪缓存的处理器。

背景技术

在处理器中所执行的指令经编码成为壹及零的序列。对于某些处理器架构，指令可以以固定的长度做编码，诸如特定数目的字节。对于其它架构，诸如 x86 架构，指令的长度可予改变。所述 x86 处理器架构定义的可变长度的指令集(意即指令集中的不同指令乃藉由不同的字节数目所定义出)，例如，80386 及 x86 处理器的后续版本使用在 1 至 15 字节之间的字节以定义特定的指令。指令具有操作码，所述操作码可以是 1 至 2 字节，并且额外的字节可以加入以定义地址模式、操作元及关于待执行的指令的额外的细节。

在某些处理器架构中，每个指令在执行之前可以译码成为一个或多个的较简单的操作。译码一个指令亦可以包含存取寄存器重新更名对映以决定在所述指令中的每个逻辑寄存器所对映的实体寄存器及 / 或以分配实体寄存器存储所述指令的结果。

指令可以部分依据在所述处理器内部所产生的分支预测而取至处理器的译码部分内。通常，所述处理器的指令取的频宽及译码部分可以决定是否所述执行核心在每一个执行周期期间是完全使用的。因此，最好能够在所述处理器的指令取及译码部分内提供足够的频宽以尽可能地保持所述执行核心完全供应有工作的状态。

发明内容

本发明揭示依据处理器在跟踪缓存内部用于处理异常指令的方法及系统的各种实施例。在某些实施例中，系统可以包含指令缓存、包含多个跟踪缓存登录的跟踪缓存及连结至所述指令缓存及所述跟踪缓存的跟踪产生器。所述跟踪产生器可用以接收藉由所述指令缓存所输

出而用于存储于所述多个跟踪缓存登录的一者内的指令群组。所述跟踪产生器可配置成检测在所述指令群组内部的额外指令及避免将所述异常指令存储于所述多个跟踪缓存登录中的相同一个跟踪缓存登录内而成为任何非异常指令。在其中一个实施例中，所述跟踪产生器可用以完全不存储所述异常指令于所述跟踪缓存内。在另一个实施例中，所述跟踪产生器可配置成存储所述异常指令于不同于所述一个或多个的其它指令的跟踪缓存登录内，并可配置成不存储具有所述异常指令的任何其它指令于所述不同的跟踪缓存登录内。所述跟踪产生器可配置用于以至少部分译码的形式把指令存储在于所述跟踪缓存内。

所述系统亦可以包含连结至退返执行指令的退返队列。所述退返队列可配置成避免从所述跟踪缓存所取的任何指令的隐退，直到包含于与所述指令相同的跟踪缓存登录内的所有的指令已可退返为止。另外，所述退返队列可配置成避免从所述跟踪缓存所取的任何指令的隐退，直到包含于在与所述指令相同的跟踪缓存登录内部的相同的存在群组的所有指令已得退返为止。

所述系统可以包含配置成监测自所述退返队列前次退返一指令后所历经的周期数目的重启逻辑。若周期的数目超过临界数目，所述重启逻辑可组构成用以清除从所述跟踪缓存所取的处理管线执行指令及重启从所述指令缓存的执行。

包含于所述系统内的调度单元可配置成发送从所述跟踪缓存所接收的指令。所述调度单元可配置成检测从所述跟踪缓存所接收的指令群组内部的不可重新执行的指令，并提供所述不可重新执行的指令的指示给所述重启逻辑。响应于所述指示，在所述不可重新执行的指令的执行之前，所述重启逻辑可配置成清除所述处理管线及重启来自所述指令缓存的执行。

所述调度单元可配置成检测从所述指令缓存所接收的指令群组内部的异常指令及把所述异常指令的指示提供给所述跟踪产生器，俾令所述跟踪产生器检测所述异常指令。所述跟踪产生器亦可配置成检测异常指令以响应于所述异常指令的一个或多个的特性(例如操作码、寻址模式等等)。

在某些实施例中，本发明的方法可以包含：跟踪产生器接收指令

群组以存储于在连结至所述跟踪产生器的跟踪缓存内部的跟踪缓存登录内；所述跟踪产生器将包含在所述指令群组中一个或多个指令存储在所述跟踪缓存登录（162）内；所述跟踪产生器检测在所述指令群组内部的异常指令；以及响应于所述检测，所述跟踪产生器不将所述异常指令存储在具有所述一个或多个指令的所述跟踪缓存登录（162）内部。

附图说明

当参酌下列详细的说明并结合下列的图式时，可较佳地了解本发明，其中：

图 1 显示处理器的一实施例。

图 2 为依据实施例的跟踪缓存的方块图。

图 3A 说明依据实施例的例示性跟踪缓存登录。

图 3B 为依据实施例可用以判别在跟踪内的每个操作归属的所述存在群组的存在编码的表格。

图 4 为依据实施例在跟踪缓存内产生用于存储的跟踪的方法的实施例的流程图。

图 5 为依据实施例执行包含异常指令的跟踪的方法的其中一实施例的流程图。

图 6 系依据实施例以显示出计算机系统的方块图。

图 7 为依据另一实施例的计算机系统的方块图。

【主要组件符号说明】

10	步骤	12	步骤
14	步骤	16	步骤
20	步骤	22	步骤
24	步骤	26	步骤
28	步骤	30	步骤
100	处理器	100a	处理器
102	退返队列	104	调度单元
106	指令缓存	108	预取单元
116	寄存器文件	118	排程

124	执行核心	126	加载存储单元
128	资料缓存	130	结果总线

132	分支预测单元	140	译码单元
160	跟踪缓存	162	登录
164	标记符	165	指令
166	操作存储单元	166A	操作存储单元
166B	操作存储单元	166C	操作存储单元
166D	操作存储单元	166E	操作存储单元
166F	操作存储单元	166G	操作存储单元
166H	操作存储单元	168	流程控制信息
170	跟踪产生器	190	重启逻辑
200	主记忆体	266	编码
400	计算机系统	402	总线桥接
404	主存储器	406	内存总线
408	图形控制器	410	绘图加速端口总线
412A	外围组件接口装置	412B	外围组件接口装置
414	外围组件接口总线	416	第二总线桥接
418	工业标准架构装置		
420	延伸工业标准架构 / 工业标准架构总线		
422	键盘及鼠标	424	中央处理器单元总线
426	显示器	428	L2 缓存
428a	L2 缓存	612A	处理节点
612B	处理节点	612C	处理节点
612D	处理节点	614A	内存
614B	内存	614C	内存
614D	内存	616A	内存控制器
616B	内存控制器	616C	内存控制器
616D	内存控制器	618A	接口逻辑
618B	接口逻辑	618C	接口逻辑
618D	接口逻辑	618E	接口逻辑
618F	接口逻辑	618G	接口逻辑
618H	接口逻辑	618I	接口逻辑
618J	接口逻辑	618K	接口逻辑

618L	接口逻辑	620A	输入 / 输出装置
620B	输入 / 输出装置	624A	线路
624B	线路	624C	线路
624D	线路	624E	线路
624F	线路	624G	线路
624H	线路		

具体实施方式

虽然本发明易于实施各种修正及替代形式，本发明的特定的实施例藉由在所述图式中的例子而作呈现并且在此将作详细描述。然而，应所述了解的是对于本发明的图式及详细说明并非意在限定本发明于所揭示的特定的形式，而相反地，本发明是在于涵括落在由所述附加的权利要求书所定义的本发明的精神及范畴之内的所有的修正、等同及替代。应所述注意的是本标题仅用于组织化的目的并且并非意在用于限定或说明所述描述或权利要求书。再者，应所述注意的是所述文字“可以(may)”在本申请案全文中是允许的含意(具潜在有、能够的意思)，而非强制的含意(意即必须(must))。所述名词“包含(include)”及所述“包含”的衍生词意指“包含，但非限定于”。所述名词“连接(connected)”意指“直接地或非直接地连接”，并且所述名词“连结(coupled)”意指“直接或非直接地连结”。

图1为包含于处理器100的一实施例内的逻辑组件的方块图。处理器100系配置成执行存储于系统内存200内的指令。很多这些指令系执行在存储于系统内存200内的数据上，所述系统内存200可以是实体分布于整个计算机系统上并且藉由一个或多个的处理器100所存取。在某些实施例中，所述处理器100可兼容于x86架构。

处理器100可以包含指令缓存106及数据缓存128。处理器100可以包含连结至所述系统内存200的预取单元108。预取单元108可以预取来自所述系统内存200以存储在指令缓存106内部的指令码。在一实施例中，预取单元108可配置成从所述系统内存200猝发程序代码进入指令缓存106内。预取单元108可以使用各种特定程序代码预取技术及算法。预取单元108亦可以取来自指令缓存106的指令及来

自跟踪缓存 160 的跟踪进入调度单元 104 内。指令可以从指令缓存 106 取以响应于在跟踪缓存 160 内遗漏的特定指令地址。同样地, 指令可以从系统内存 200 取以响应于在指令缓存 106 内遗漏的特定地址。

调度单元 104 可配置成接收来自指令缓存 106 的指令及可配置成接收来自跟踪缓存 160 的译码及 / 或部分译码的指令。所述调度单元 104 可以包含译码单元 140 以译码来自指令缓存 106 所接收的指令。所述调度单元 104 亦可以包含当处理微码指令时而使用的微码单元。

所述调度单元 104 系配置成发送译码的指令给排程 118。一个或多个的排程 118 可以经由连结以接收来自调度单元 104 所发送的指令及发出指令给一个或多个的执行核心 124。执行核心 124 可以包含配置成执行存取数据缓存 128 的加载存储单元 126。由执行核心 124 所产生的结果可以输出至结果总线 130。这些结果可以使用作为用于后续发出指令的操作元数值及 / 或可以存储至寄存器文件 116。退返队列 102 可以连结至排程 118 及调度单元 104。所述退返队列可配置成决定每个发出的指令何时可以隐退。

指令缓存 106 在由调度单元 104 收到指令之前可以暂时地存储指令。指令码藉由从所述系统内存 200 透过预取单元 108 预取程序代码而可以提供给指令缓存 106。指令缓存 106 可以以各种配置(例如集合联合、完全联合或直接对映)而实现。

如同在此所使用的, 所述名词指令通常可以意指未译码的、部分译码的及完全译码的指令。部分及完全译码的指令亦可以称作为操作。例如, 未译码的指令可以藉由译码单元 140 描述作为受到译码成为一个或多个的可直接在执行核心 124 内执行的组件操作。调度单元 104 可以输出包含可藉由所述执行核心 124 执行的位编码操作的讯号以及操作元地址信息、实时数据及 / 或位移资料。当接收包含寄存器的更新的指令时, 所述调度单元 104 可以保留在寄存器文件 116 内部的寄存器位置以存储推测寄存器状态(在另一个实施例中, 重新定序缓冲器可以使用以存储一个或多个的用于每个寄存器的推测寄存器状态)。寄存器对映可以转译来源及目的操作元的逻辑寄存器名称成为实体寄存器名称以促使寄存器重新命名。此类的寄存器对映可以跟踪在寄存器文件 116 内部的寄存器何者目前为已分派或未分派。

当藉指令由调度单元 104 所处理时, 若所需的操作元为寄存器位置, 寄存器地址信息可以递送至寄存器对映或重新定序缓冲区。例如, 在所述 x86 架构中, 具有八个 32 位逻辑寄存器(例如 EAX、EBX、ECX、EDX、EBP、ESI、EDI 及 ESP)。实体寄存器文件 116(或重新定序缓冲区)包含用于改变这些逻辑寄存器的内容的结果的存储, 而允许脱序执行。在寄存器文件 116 内的实体寄存器可以分派以存储修正其中一个所述逻辑寄存器的内容的每个指令的结果。因此, 在特定程序的执行期间的各种位置处, 寄存器文件 116(或者在另一个实施例中, 重新定序缓冲区)可以具有一个或多个含有给定逻辑寄存器的所述推测执行的内容的寄存器。

寄存器对映可以指定实体寄存器给定义成用于指令的目的操作元的特定的逻辑寄存器。寄存器文件 116 可以具有指定给定义成为在给定的指令中的来源操作元的逻辑寄存器的一个或多个的预先分配的实体寄存器。所述寄存器对映可以提供标记符给最新指定给所述逻辑寄存器的实体寄存器。此标记符可以使用于存取在所述寄存器文件 116 内的所述操作元的资料数值或用于接收经由传递在所述结果总线 130 之上的结果的所述资料数值。若所述操作元符合内存位置, 所述操作元数值可以透过加载存储单元 126 而提供在所述结果总线(用于传递及 / 或存储于寄存器文件 116 之内的结果)之上。当所述指令藉由其中一个所述排程 118 发出时, 操作元资料数值可以提供给执行核心 124。需要注意的是在另一个实施例中, 当指令发送时, 操作元数值可以提供给对应的排程 118(而非当所述指令发出时, 提供给对应的执行核心 124)。

图 1 的所述处理器 100 支持脱序执行。退返队列 102(或者另外, 重新定序缓冲区)可以保持用于寄存器读取及写入操作的所述原始程序顺序的跟踪、允许推测指令执行及分支错误预测回复及促进精确排除。在很多实施例中, 退返队列 102 可以产生类似于重新定序缓冲区的功能。然而, 不像典型的重新定序缓冲区, 退返队列 102 可以不提供任何资料数值存储。在另一个实施例中, 退返队列 102 可以产生更加类似于重新定序缓冲区的功能并且亦藉由提供用于推测寄存器状态的存储的资料数值而支持寄存器重新命名。在某些实施例中, 退返队列 102

可以以先入先出配置方式而实现,其中当操作为有效时,所述操作移至所述缓冲区的“底部(bottom)”,因此在所述队列的“顶部(top)”处产生用于新的登录的空间。当指令隐退时,退返队列 102 可以在寄存器文件 116 内解除分配寄存器,所述寄存器遂不再需要存储推测寄存器状态,所述退返队列 102 然后提供讯号给寄存器对映而显示那一个寄存器目前是闲置的。藉由维持在寄存器文件 116 内部的推测寄存器状态(或者,在另一个实施例中,在重新定序缓冲区的内部)直到产生那些状态的指令是有效的,若分支预测是不正确的,依循错误预测的路径的推测执行的指令结果在所述寄存器文件 116 内可以是无效的。

提供在调度单元 104 的输出处的所述位编码操作及实时数据可以传递至一个或多个的排程 118。应所述注意如同在此所使用的,排程系用以检测何时指令已待执行及发出实时指令给一个或多个的执行单元。例如,保留站(reservation station)为排程的一种形式。每个排程 118 可以能够保留指令信息(例如,位编码的执行位以及操作元数值、操作元标记符及 / 或实时数据)用于数个等待发出至执行核心 124 的待决指令。在某些实施例中,每个排程 118 可以不提供操作元数值存储。相反地,每个排程可以监测发出的指令及在寄存器文件 116 内既有的结果以决定何时操作元数值将可藉由执行核心 124(从寄存器文件 116 或结果总线 130)读取。在某些实施例中,每个排程 118 可以与预设的功能性单元(例如整数单元、浮点单元、加载存储单元等等)及 / 或执行核心 124 结合。在其它实施例中,单一排程 118 可以发出指令给超过一个的接收器。

排程 118 可暂时存储藉由所述执行核心 124 所执行的操作信息。指令发出至执行核心 124 以供执行,俾响应任何需要的可实时获得以供执行的操作元的数值。因此,执行指令的顺序可以不同于所述原始程序指令序列的顺序。

在一实施例中,每个所述执行核心 124 可以包含功能性单元,诸如用以执行加法及减法的整数算术操作以及转移、旋转、逻辑操作及分支操作。浮点单元亦可以包含在内以供浮点操作。一或多个的所述执行核心 124 可配置成执行用于藉由加载存储单元 126 所执行的加载及存储记忆体操作的地址产生。

所述执行核心 124 亦可以提供关于条件分支指令的执行的执行的信息给分支预测单元 132。若来自所述执行核心 124 的信息显示分支预测是不正确，所述分支预测单元 132 可以清除接续于已经进入所述指令处理管线的错误分支的指令并且重新指向预取单元 108。所述重新指向预取单元 108 接着可以从指令缓存 106、及 / 或系统内存 200 开始取所述正确的指令集。在此种状况下，在所述错误分支指令之后所发生的所述原始程序序列内的指令的结果可以舍弃，包含推测执行及暂时存储于加载存储单元 126 及 / 或寄存器文件 116 内的结果。

若更新寄存器数值，在执行核心 124 内的组件所产生的结果可以输出在所述结果总线 130 之上而至所述寄存器文件 116。若更改内存位置的内容，在执行核心 124 内部所产生的结果可以提供给所述加载存储单元 126。

加载存储单元 126 提供接口于执行核心 124 及资料缓存 128 之间。在一实施例中，加载存储单元 126 可与一加载存储缓冲区配置，所述加载存储缓冲区具有数个用于待决加载或存储的资料及地址信息的存储位置。所述加载存储单元 126 亦可以执行用于加载指令对照待决存储指令的附属性确认以确保数据一致性的维持。

数据缓存 128 为用以暂时存储在加载存储单元 126 及所述系统内存 200 之间转移的资料的缓存内存。类似于上文所描述的所述指令缓存 106，所述数据缓存 128 可以以各种特定的内存配置而实现，包含集合联合配置。此外，数据缓存 128 及指令缓存 106 在某些实施例中可于单一化的缓存中实现。

退返队列 102 可以提供确认程序跟踪的讯号予跟踪产生器 170。跟踪产生器 170 亦可为填覆单元。跟踪产生器 170 可将藉由退返队列 102 所确认的跟踪存入跟踪缓存 160 内。每个跟踪可以包含存储在顺序而非程序顺序内的指令并且所述指令为部分数个不同的基本区块。基本区块为所有指令将会执行或所有指令将不执行的指令集。换言之，若任何在基本区块内的指令执行时，在所述基本区块内的所有所述其余的指令亦将会执行。基本区块的例子为刚好在分支指令之后开始及随着另一个分支指令而结束。存储在跟踪缓存 160 之内的跟踪在某些实施例中可以包含译码或部分译码的指令。如同在此所使用的，“跟踪”

为存储在所述跟踪缓存 160 内的单一跟踪缓存登录(例如单一系列或线路)内部的指令群组。在登录内的所有信息可以在同一时间存取(例如响应于跟踪读取周期的宣告)。应所述注意的是在另一实施例中,跟踪产生器 170 可以连结至所述处理器的前端(例如在所述调度单元之前或之后)及用以产生跟踪以响应于所取及 / 或发送的指令。

预取单元 108 可以从跟踪缓存 160 取指令进入调度单元 104。从跟踪缓存 160 取指令,相对于从指令缓存 106 取,可以提供跨越分支边界的改良的取效能。例如,若分支是经预测而予采用,在所述预测采用路径内的指令可已存储在如同于跟踪缓存 160 的分支内部的相同的跟踪内。此外,在指令是以至少部分译码形式存储在跟踪缓存 160 中的实施例内,来自所述跟踪缓存的取指令可允许所述译码单元 140 成为至少部分旁通,以使用于所述缓存指令的调度单元的数目减少。因此,当执行跟踪超过一次时,所述跟踪缓存 160 可使用以部分(或完全地)译码所述缓存指令所耗时间摊列于数个执行迭代。

应所述注意的是除了在此所呈现的组件之外,处理器 100 可以包含及 / 或连结至很多其它的组件。例如,额外的缓存阶层可以包含(内部及 / 或外部连结至处理器 100)在处理器 100 及系统内存 200 之间。同样地,处理器 100 在某些实施例中可以包含配置成控制系统内存 200 的整合的内存控制器。此外,在逻辑组件之间之内连接在实施例之间可以作变化。

跟踪缓存

图 2 依据一实施例说明例示性的跟踪缓存 160。所述跟踪缓存 160 可以包含数个登录 162。每个登录 162 存储包含数个指令 165 的跟踪。在跟踪内的所述指令 165 可以不存储于程序顺序内。例如,当采用所述分支时(相对于依循在程序顺序内的分支的指令),登录 162 可以存储分支指令及为所述分支的目的指令。在某些实施例中,所述指令 165 在每个跟踪缓存登录 162 内可以以至少部分译码的形式而存储。如同在此所使用的,所述名词“跟踪”是用于意指存储在单一跟踪缓存登录 162 内的一个或多个的指令的群组。

返回参阅图 1,应所述注意的是从所述跟踪缓存 160 所取的指令流的处理可以不同于从指令缓存 106 所取的指令流。来自指令缓存 106

的指令流可以包含确认指令边界的信息，即使在指令译码成为一个或多个的组件操作之后。这种边界信息可以藉由退返队列 102 所用以决定对应于特定指令的操作何时可以隐退。通常，所述退返队列 102 以原始的程序顺序操作以退返操作。对应于所述相同的指令的操作可以在相同的时间隐退。

相反地，若所述指令在存储于跟踪缓存 160 内之前已经译码及 / 或修正(例如藉由结合指令以增加在跟踪缓存 160 的内部的存储效率)，来自跟踪缓存 160 的指令流可以不包含确认指令边界的信息。因此，退返队列 102 可须退返包含在运行表面上的此类指令流内的指令。例如，在某些实施例中，退返队列 102 可以不退返包含在跟踪内的任何指令，直到在所述跟踪内的所有指令得予退返。

在某些跟踪缓存型处理器之内可能产生的其中一项问题在于某些指令，在此称为“特殊”指令，可能需要特别的处理，所述处理影响所述些指令何时可以退返。由于在所述相同的跟踪中如同异常指令的其它指令(包含在程序顺序中较早产生的指令)的退返可以视准备好退返的所述异常指令而定，此类异常指令可以阻挡在所述跟踪中的其它指令的退返。在某些情况下，若所述异常指令的不退返，则可阻挡包含于所述相同跟踪内的较旧的操作的退返。此外，若异常指令由于缺乏来自所述跟踪缓存所提供的指令边界信息而阻挡在所述相同的跟踪内的其它指令的退返，在所述跟踪内的所有所述非可退返的指令(及在程序顺序中，已经准备好发送的任何较新的指令)可能必须从所述指令缓存重新取并且重新执行。

异常指令包含产生特例或中断的指令。异常指令相对于非异常指令可能会花费明显较长时间来完成。例如，执行核心 124 在一至三个执行周期内可以完成典型的非异常指令。相反地，异常指令可能须 10 倍(或更多)于用以执行的执行周期的时间来产生中断。具有至少某些异常指令，对所有较特定异常指令为旧的所述指令(在程序顺序中)能够独立于所述异常指令而退返可能是有益的或甚至是需要的。如同上文所提及的，在跟踪中缺乏指令边界信息可能减少或失去不同的指令于所述相同的跟踪中独立退返的能力。因此，在跟踪中的异常指令包含于其它指令可能最后造成在所述跟踪中的所有所述指令必须从指令缓存

重新取以获得所述边界信息。

为了减少异常指令阻挡其余指令的退返及 / 或造成其它指令舍弃、重新取及重新执行的可能性,跟踪产生器 170 配置成检测异常指令及不存储在包含任何其它指令的跟踪内的此类指令。在某些实施例中,跟踪产生器 170 可以存储在仅包含所述异常指令的跟踪中所检测到的异常指令。虽然此类配置可能降低跟踪缓存存储效率,然其得从所述跟踪缓存继续取而非造成在所述指令缓存 106 及跟踪缓存 160 之间取的切换。在其它实施例中,跟踪产生器 170 可以完全不存储在跟踪缓存 160 内所检测到的异常指令,造成所述异常指令一直从指令缓存 106 取。跟踪产生器 170 为用于在跟踪缓存内部的跟踪缓存登录中存储指令群组的工具的例子。

跟踪产生器 170 藉由检验用于存储在所述跟踪缓存内的每个候选指令的特定操作码、操作元、地址模式及其它特性而可以检测出异常指令。在一实施例中,确认潜在异常指令的特性可以程序化于所述跟踪产生器 170 之内。需注意的是在所述等实施例中,所述跟踪产生器 170 可配置成经常性地确认特定形式的指令为异常指令,即使那些形式的指令在特定环境下可能是可执行的而不需要特别的处理。在某些实施例中,译码单元 140 在指令译码期间可以产生确认异常指令的标记符。每个标记符可以透过所述处理器 100 结合由所述标记符确认成为异常的译码指令而递送。除能(或者除了)独立地决定译码的指令是否特殊之外,跟踪产生器 170 可视藉由译码单元 140 所产生的标记符而定以确认异常指令。因此,跟踪产生器 170 及译码单元 140 两者为用于检测在所述指令群组内部的异常指令的工具的例子。

响应于检测存储在所述跟踪缓存 160 中的指令流内的异常指令,跟踪产生器 170 可以结束指令存储于所述目前跟踪内(即使在所述跟踪内仍有存储额外指令的空间)。若跟踪产生器 170 配置成在跟踪缓存 160 内的独立跟踪中存储异常指令,所述跟踪产生器可以分配新的跟踪缓存登录以存储所述异常指令。所述跟踪产生器 170 可以存储下一个指令,所述指令刚好在于目前另一个跟踪缓存登录内的程序顺序中的异常指令之后产生。若所述跟踪产生器 170 为相反地配置成从所述跟踪缓存 160 排除全部异常指令,当检测到所述异常指令时(除非未有其它

旧的指令已经准备好存储于所述目前跟踪内)所述跟踪产生器可以仅结束所述目前跟踪并且开始存储于程序顺序中的所述下一个非异常指令,所述指令为用于存储在所述下一个跟踪的跟踪缓存内的候选指令。需注意的是所述跟踪产生器 170 可以设定流程信息于所述目前跟踪内以指示所述异常指令为所述下一个欲取的指令。因此,响应于所取的跟踪,预取单元 170 可配置成从指令缓存 106 预取在所述流程信息内所确认的指令。因此,由于所述异常指令系从所述跟踪缓存 160 排除,所述执行串行将从所述跟踪缓存 160 切换至所述指令缓存 106。

某些异常指令不一定能为跟踪产生器 170 所检测到。例如,某些指令每一次执行时,其行为可能是不同的。因为这些指令的特性(例如此类指令甚少是异常的),跟踪产生器 170 可配置成被动地确认指令为特殊者。因此,这些异常指令可以藉由跟踪产生器 170 与其它指令存储在跟踪内。由于在所述相同的跟踪内的其它指令的阻挡的退返,为了避免此类异常指令的排除造成问题,重启逻辑 190 可以监测退返队列 102 前一次退返任何操作后,已经过多少阻挡周期。若没有操作在时脉周期的预定的数目内已经退返,重启逻辑 190 可以从指令缓存 106 重启处理器 100 的执行。在一实施例中,重启逻辑 190 可以藉由清除所述处理管线及藉由重新取在处理器 100 内部与所述最旧待决的非退返指令开始的指令而重启执行。当所述指令从指令缓存 106 取时,关于指令边界的信息将可提供予退返队列 102。因此,若藉由重启逻辑 190 所侦到的所述非退返是藉由阻挡跟踪的退返的异常指令所造成,从指令缓存 106 取所述指令可以允许所述退返队列 102 独立地退返所述有问题的异常指令而不需要非必要地阻挡其它指令的退返。

某些指令可以是不可重新执行的,并且若所述指令从跟踪缓存 160 而执行、藉由包含在所述相同的跟踪内的异常指令的退返所阻挡以及接着因为重启逻辑 190 的操作而重新执行,此类的指令可能是不正确的。例如,特定负载(例如某些输入/输出负载)的效能可以修正所读取的数值。若此类指令重新执行,为了避免可能造成的问题,跟踪产生器 170 可配置成确认这些指令为不与其它指令存储在跟踪内的异常指令的另一个类别。因此,所述跟踪产生器 170 可以从所述跟踪缓存 160 排除这些异常指令或者存储这些异常指令于不包含任何其它指令

(例如一般如同上文所描述关于异常指令)的单独跟踪内。

如同某些其它异常指令，一些非可重新执行的指令不一定能为跟踪产生器 170 所检测出。例如，加载在其中一个程序重复及输入 / 输出寄存器所述下一个重复期间可以存取系统内存 200 地址。在所述第一重复之后，所述加载可以存储于跟踪内。在所述下一个重复期间，基于所述输入 / 输出寄存器读取的特性所述加载可以是非可重新执行的。因此，若在所述相同跟踪内作为所述加载的另一个指令是异常的(例如，若另一个指令产生页面错误)，所述跟踪的退返可以延迟足够长的时间以触发重启逻辑 190。然而，这可能令人无法接受地造成所述加载的重新执行。为了避免这种情况，调度单元 104 或处理器 100 内部的另一个组件可配置成确认在所述加载实际地执行之前所述加载在特定时间点成为已非可重新执行的。响应于在从跟踪缓存 160 所取的指令群组内部的非可重新执行的指令的检测，重启逻辑 190 可造成所述处理器管线受到清除(在所述非可重新执行的指令的执行之前)并且从指令缓存 106 重启执行。

例示性跟踪

图 3A 说明配置成存储至少部分译码的指令的跟踪缓存登录 162 的一实施例。在所述图 3A 的实施例中，跟踪缓存登录 162 可以存储多达包含于操作存储单元 166A 至 166H 内的译码及 / 或部分译码的指令群组中的八个组件操作。需要注意的是其它实施例可以包含较少或额外的操作存储单元 166，而允许在每个跟踪缓存登录 162 内部的操作的不同的数目的存储。

在跟踪缓存登录 162 内部的某些操作存储单元 166 可以保留予某些形式的操作。例如，在一实施例中，操作存储单元 166A 至 166D 可以使用于存储记忆体操作。同样地，操作存储单元 166E 至 166H 可以使用于存储资料操作。需要注意的是其它实施例可以不同地结合某些形式的操作与某些操作存储单元(或者并非全部)。

除了包含数个操作存储单元 166 之外，每个跟踪缓存登录 162 亦包含确认标记符 164 及流程控制信息 168。标记符 164 可以类似于在指令缓存 106 内的标记符，允许预取单元 108 决定给定的操作于跟踪缓存 160 内是否是正确或错误的。例如，标记符 164 可以包含所有或某些所

述地址位而确认在所述跟踪缓存登录内部的指令(例如在程序序顺中, 所述标记符可以包含存储在所述跟踪内的所述最早指令的地址)。在某些实施例中, 所述标记符可以包含足够的信息, 所述信息乃指每个指令的地址(或者至少在每个存在群组内部的所述最早的指令, 于下文中作更详细的描述)可以使用存储在所述跟踪内的信息而独立地确认者。在其它实施例中, 只有在跟踪内的最早的指令的地址可以获得。

在所述的实施例中, 每个跟踪可以包含多达两个的分支指令。其它实施例可以包含较少或额外的分支指令于每个跟踪内。流程控制信息 168 可以包含用于包含在所述跟踪内的每个分支指令的标记。视个别的分支的分辨率(采用, 不采用)而定, 所述标记可以是确认控制所应分支至的地址的指示。因此, 流程控制信息 168 的每个项目可以结合特定的分支操作。例如, 在一实施例中, 在跟踪内的一个流程控制信息存储位置可以结合在所述跟踪内的第一分支操作, 且其它流程控制信息存储位置可以结合在所述跟踪内的第二分支。另外, 所述流程控制信息可以包含确认所述流程控制信息所结合的所述分支操作的标记符或其它信息。在另一实施例中, 确认那一个流程控制信息对应于分支操作的分支预测及 / 或信息可以与在操作存储 166 内的所述分支操作一起存储。

在跟踪内提供增加的退返粒度(Granularity)

如同上文所提及的, 在跟踪内关于指令边界的信息是有限的。例如, 在存储于跟踪内之前, 若指令是部分译码成为所述指令本身的组件操作, 在所述跟踪内缺乏描述所述不同的指令的信息可以包含于所述跟踪内。再者, 若在译码之后, 不同的指令的组件操作的组合、重新定序及 / 或修正, 可能使确认指令边界更为困难。

为了在跟踪内提供增加的指令边界信息, 在跟踪内的译码指令可以再细分成为不同的存在群组。跟踪产生器 170 可以标记存储在特定跟踪缓存登录内部的每个译码的指令成为属于在所述跟踪内的特定存在群组。关于在每个存在群组内的经译码的指令的指令边界的信息可能无法获得, 然而, 所述存在群组本身可以创造使得所述译码指令可以在存在群组边界上退返。

在跟踪内的每个操作 166 可以具有相关联的存在指示。每一操作

的存在指示可以与本身相结合的操作在其中一个操作存储单元 166 内一起存储。存在指示系确认出每个操作所属的存在群组。每个存在群组为在所述跟踪内的操作群组，所述跟踪为部分所述相同的基本区块。例如，在程序顺序中，在所述跟踪内的达到及包含所述第一分支操作的操作可以包含于其中一个存在群组内。本身的执行依赖所述第一分支操作的分辨率的操作可以包含于另一个存在群组之内。因此，存在群组类似于基本区块，因在所述相同的存在群组内的操作将全部执行或者全部不执行。存在群组不同于基本区块，因在所述相同的基本区块内的操作可以包含于不同的跟踪内(例如在所述基本区块内的某些操作可以包含于其中一个跟踪内的最终存在群组内并且在所述基本区块内的剩余操作可以包含于另一个跟踪内的所述第一存在群组中)。因此，在所述相同的存在群组内的操作必须是所述相同的基本区块的一部分，但是在所述相同的基本区块内的操作可以不需要是所述相同的存在群组的一部分(意即若所述基本区块超过多个的跟踪缓存登录)。

所述存在指示确认相对于包含在所述相同的跟踪缓存登录内的所述分支操作的每一个操作的程序顺序。在第一存在群组内的操作并非视在所述跟踪内的第一分支操作的结果而定。需要注意的是由于所述第一分支操作非有条件的执行，所述第一分支操作作为所述第一存在群组的一部分。在所述第一存在群组内的操作可以以不同于经由使用以确认发生在所述第一分支操作之后的操作的第二存在群组的存在指示而确认。同样地，视所述第一分支操作及第二分支操作(意即包含于所述第二存在群组内的所述最终操作)两者的结果而定的位在第三存在群组内的指令，可以以另一个存在指示而确认。存在指示允许操作以不同于本身的程序顺序的顺序而存储在跟踪缓存登录 162 内部，同时仍决定出所述操作的程序顺序(相对于所述分支操作)。

此外，存在指示允许调度单元 104 预测在特定跟踪内的操作何者会执行。在所述第一存在群组内的操作(意即非视包含在跟踪内的第一分支而定的操作)将一直执行(在所述跟踪执行内的任何操作)。然而，在所述第二存在群组内的操作的执行(意即视所述第一分支而定的操作)视所述第一分支的结果而定。例如，在其中所述跟踪产生器 170 包含在处理器 100 的处理管线之后段处的实施例，操作可依据操作执行

的顺序而存储于跟踪内，以便若所述操作随后再次执行，所述操作可以从所述跟踪缓存而非从所述指令缓存所存取。因此，若存储在所述跟踪内的第一分支的执行为所述第一次分支的执行，则在所述第一分支已采用时，存储在第二存在群组内的所述操作可以是应所述执行的操作。因此，若所述跟踪稍后提供给调度单元 104 并且用于所述第一分支的目前的分支预测“未采用”，所述调度单元 104 可以预测在所述第二存在群组内的操作不应所述执行。若所述第一分支并未采用，结合所述第一分支操作的所述流程控制信息 168 亦可开始预取应所述执行的指令或跟踪。

图 3B 显示可以使用于一实施例中的例示性的存在编码。所述第一存在编码，“无条件存在”，可用以确认在所述跟踪内的所述第一(在程序顺序中)存在群组。在这个存在群组内的操作是无条件存在，前题系若执行所述跟踪，这些操作将一直执行。所述第一存在群组可以包含分支操作成为在所述存在群组内的最终操作(在程序顺序中)。这些操作非视包含在所述相同的跟踪内的任何分支操作而定。在很多实施例中，这些操作相对于彼此在所述跟踪内可以以任何顺序而存储。

所述次一个存在编码，“接续于第一分支”是用于确认在所述跟踪内的所述第二(在程序顺序中)存在群组。这些操作是视所述第一分支操作的结果而定，所述第一分支操作包含于前一个存在群组内。需要注意的是若非推测执行在所述存在群组内的任何操作，在所述存在群组内的所有指令将会执行。

所述第三存在编码，“接续于第二分支”是用于确认在所述跟踪内的第三(在程序顺序中)基本区块。这些操作是视所述跟踪内的所述第一及所述第二分支操作两者的结果而定。因此，即使执行具有所述第二存在编码的操作，这些操作可以不执行。如同上述，若非推测执行所述存在群组内的任何操作，在所述存在群组内的所有操作将会执行。

所述“无条件停滞(unconditionally dead)”存在编码可用以确认未使用的操作存储单元 166。若超过分支操作的最大数目发生在操作集合内，可不使用操作存储单元 166。例如，若多达八个操作可以存储在跟踪内并且若仅有两个分支操作可以存储在跟踪内，则若给定的八个操作集合包含三个分支操作时，所述跟踪可存储少于八个操作。因此，

若最大限制系设在可以存储在跟踪内的分支操作的数目 N ，在所述跟踪内的所述第 N 个分支操作的存储可以结束所述跟踪。再者，在某些实施例中，若特定操作存储在跟踪内，即使操作可予存储，后续的操作(在程序顺序中)可以不存储在所述跟踪内。例如，子过程调用操作可以结束跟踪(例如，以在所述子程序内部的第一操作可以存储成为在另一个跟踪内的第一操作，使所述操作的标记符成为跟踪标记符)。同样地，子程序回传操作可以结束跟踪。

再参照图 1，跟踪产生器 170 可配置成产生每个操作的适当的存在编码，并且用以存储在所述相同的操作存储单元 166 内的所述存在编码而成为在跟踪缓存 160 中的跟踪缓存登录 162 内的操作。依据关于所述每一分支操作首次执行的每个分支的结果的信息，跟踪产生器 170 可以产生用于每个操作的存在编码。这种信息可以由退返队列 102 而提供。额外的信息可以由分支预测单元 132 所提供而确认用于在一个或多个的先前执行内的每个分支的每个预测的预测的结果及 / 或成功。

当指令从跟踪缓存 160 取时，与每个操作结合的所述存在编码可与所述指令经由所述处理器传送。所述退返队列 102 可以使用用于每个操作的存在编码以决定指令可以退返于操作上的边界。例如，除了替阻挡在跟踪内的操作的退返而所述跟踪内的任何所有的操作皆可退返外，一旦当在所述跟踪内的相同的存在群组中的所有所述指令准备好退返时，所述退返队列 102 可以退返在所述跟踪内的所述操作的子集合。在某些实施例中，这种增加的退返粒度可以减少重启逻辑 190 由于阻挡退返所需要重启执行的可能性。

图 4 说明产生用于存储在跟踪缓存内存储的跟踪的方法的一实施例。在步骤 10，接收一个或多个的指令而存储在所述跟踪缓存内。若任何所述指令是异常的(例如若任何指令可能造成例外或者若任何指令为不能够重新执行者)，所述异常指令可如同任何其它的指令不存储在所述相同的跟踪内，如同在步骤 12 及步骤 14 所示者。若在步骤 10 所接收的指令均非异常的，这些指令可以一起存储在所述相同的跟踪内及 / 或包含在所述相同的跟踪内而成为额外非异常指令，如同步骤 12 及 16 处所示者。

图 5 为处理错误地包含在所述跟踪缓存内部的跟踪内的未检测到的异常指令的方法的一实施例的流程图。在步骤 20, 指令流从所述跟踪缓存取。若异常指令在所取的指令流内部检测到, 如同在步骤 22 所决定的, 所述管线可予清除并且处理器执行可以由所述指令缓存重启, 如同在步骤 24 所示者。

在所述指令流内部的某些异常指令在步骤 22 可能检测不到。若此类的指令造成例外, 所述例外接而会如同异常指令阻挡在所述相同的跟踪内的其它指令的退返, 所述处理器管线可予清除并且执行可以由所述指令缓存重启, 如同在步骤 26 及 28 所示者。决定是否阻挡指令的退返可以包含检测指令退返。若为期一预定数目的周期没有指令退返, 可以确认退返受到阻挡。若指令退返并未受到阻挡, 从所述跟踪缓存所取的所述指令流可予执行及退返, 如同在步骤 26 及 30 所示者。

例示性计算机系统

图 6 显示包含透过总线桥接 402 连结至各种系统组件的处理器 100 的计算机系统 400 的一实施例的方块图。处理器 100 可以包含如同上文所描述的调度单元 104、跟踪缓存 160、退返队列 102、重启逻辑 190 及 / 或跟踪缓存产生器 170。计算机系统的其它实施例是可能且能推及的。在所述描述的系统, 主存储器 200 透过内存总线 406 连结至总线桥接 402, 并且图形控制器 408 透过绘图加速端口总线 410 连结至总线桥接 402。数个外围组件接口装置 412A 至 412B 透过外围组件接口总线 414 连结至总线桥接 402。第二总线桥接 416 亦可供一电子介面透过延伸工业标准架构 / 工业标准架构总线(EISA / ISA bus)420 而兼容于一个或多个的延伸工业标准架构或工业标准架构装置 418。在这个例子中, 处理器 100 透过中央处理单元总线 424 连结至总线桥接 402 及连结至选择的 L2 缓存 428。在某些实施例中, 所述处理器 100 可以包含整合的 L1 缓存(未图标)。

总线桥接 402 提供接口于处理器 100、主存储器 404、图形控制器 408 及连结至外围组件接口总线 414 的装置之间。当操作从其中一个连接至总线桥接 402 的装置所接收时, 总线桥接 402 确认所述操作的目标(例如特定的装置, 或者在外围组件接口总线 414 的例子中, 所述目标是在外围组件接口总线 414 之上的装置)。总线桥接 402 递送所述操

作给所述目标装置。总线桥接 402 通常从藉由所述来源装置或总线所使用的协议转译操作给藉由所述目标装置或总线所使用的协议。

除了提供接口至用于外围组件接口总线 414 的工业标准架构 / 延伸工业标准架构总线之外, 第二总线桥接 416 可以并入额外的功能性。输入 / 输出控制器(未图标), 不论是从第二总线桥接 416 外部或与第二总线桥接 416 整合, 亦可以包含在计算机系统 400 内部, 以提供操作支持给键盘及鼠标 422 及各种序列及并行端口。外部缓存单元(未图标)在其它实施例中亦可以连结至在处理器 100 及总线桥接 402 之间的中央处理器单元总线 424。另外, 所述外部缓存可以连结至总线桥接 402 并且用于所述外部缓存的缓存控制逻辑可以整合至总线桥接 402 之内。L2 缓存 428 系示为所述处理器 100 的背面组构。需要注意的是 L2 缓存 428 可以与处理器 100 分离, 与处理器 100 整合至卡匣(例如插槽 1 或插槽 A)内, 或者甚至与处理器 100 整合至半导体基板之上。

主存储器 200 为存储应用程序及处理器 100 主要由其所执行的内存。适当的主存储器 200 可以包含动态随机存取内存(Dynamic Random Access Memory, DRAM)。例如, 复数组同步动态随机存取内存(Synchronous Dynamic Random Access Memory, SDRAM)或 Rambus 动态随机存取内存(Rambus Dynamic Random Access Memory, RDRAM)均可适用。

外围组件装置 412A 至 412B 系用以说明各种外围装置, 诸如网络适配卡、视讯加速器、音讯卡、硬式或软式磁盘驱动器或驱动控制器、小型计算机系统接口(Small Computer Systems Interface, SCSI)转接器及电话卡。同样地, 工业标准架构装置 418 系用以说明各种形式的外围装置, 诸如调制解调器、声卡及诸如通用接口总线(BPIB)或字段总线适配卡(field bus interface cards)的各种资料撷取卡。

图形控制器 408 系用以控制在显示器 426 上的文字及影像的呈现。图形控制器 408 可包括一般在所述技艺中已知的典型的绘图加速器, 以呈现能有效地移转至主存储器 200 内及由主存储器 200 移转出的三维数据结构。图形控制器 408 因此可以是绘图加速端口总线 410 的主控者, 因所述图形控制器 408 能请求及接收存取在总线桥接 402 内部的目标接口, 藉以取得主存储器 200 的存取。所成的绘图总线系供应

从主存储器 404 而来的资料的快速取回。对于特定操作, 图形控制器 408 更可配置成产生外围组件接口协议处理于绘图加速端口总线 410 之上。总线桥接 402 的绘图加速端口接口可以因此包含功能性支持绘图加速端口协议处理以及外围组件接口目标及启始器处理两者的功能。显示器 426 为影像或文字能呈现在其上的任何电子显示器。适当的显示器 426 包含阴极射线管(“cathode ray tube, CRT”)、液晶显示器(“liquid crystal display, LCD”)等等。

应所述注意的是, 虽然所述绘图加速端口、外围组件接口及工业标准架构或延伸工业标准架构总线已做为所述的例子, 任何总线架构可以依所需而替换。更应所述注意的是计算机系统 400 可以是包含额外的处理器的多任务处理的计算机系统(例如显示为计算机系统 400 的选择性的组件的处理器 100a)。处理器 100a 可以类似于处理器 100。尤其, 处理器 100a 在一实施例中可以是处理器 100 的相同复制。处理器 100a 经由独立的总线(如同在图 6 中所示者)可以连接至总线桥接 402 或者可以与处理器 100 分享中央处理器单元总线 224。再者, 处理器 100a 可以连结至类似于 L2 缓存 428 的选择的 L2 缓存 428a。

现参照图 7, 所述图显示计算机系统 400 的另一实施例。其它的实施例是可能并可推及的。在图 7 的实施例中, 计算机系统 400 包含数个处理节点 612A、612B、612C 及 612D。每个处理节点经由包含在每个个别的处理节点 612A 至 612D 内部的内存控制器 616A 至 616D 而连结至相对的内存 614A 至 614D。此外, 处理节点 612A 至 612D 包含用以在所述处理节点 612A 至 612D 之间通讯的接口逻辑。例如, 处理节点 612A 包含用于与处理节点 612B 通讯的接口逻辑 618A、用于与处理节点 612C 通讯的接口逻辑 618B 及用于与另一个处理节点通讯的第三接口逻辑 618C(未图标)。同样地, 处理节点 612B 包含接口逻辑 618D、618E 及 618F; 处理节点 612C 包含接口逻辑 618G、618H 及 618I; 及处理节点 612D 包含接口逻辑 618J、618K 及 618L。处理节点 612D 经由接口逻辑 618L 而连结以与多个输入 / 输出装置通讯(例如在菊链式串接组构中的装置 620A 至 620B)。其它的处理节点可以与其它输入 / 输出装置以类似的方式而产生通讯。

处理节点 612A 至 612D 系实现用于内部处理节点通讯的封包型连

结。在本实施例中，所述连结以单向线路组而实现(例如，线路 624A 用以从处理节点 612A 传送封包至处理节点 612B 并且线路 624B 用以从处理节点 612B 传送封包至处理节点 612A)。其它组线路 624C 至 624H 系用以传送封包于其它处理节点之间，如同在图 7 中的说明。通常，每组线路 624 可以包含一个或多个的资料线路、对应于所述资料线路的一个或多个的时脉线路及显示所传送的封包类型的一个或多个的控制线路。所述连结可以以缓存耦合的方式运作而用于处理节点间的通讯或者以非耦合的方式而用于处理节点及输入 / 输出装置(或总线桥接至习知架构的输入 / 输出总线，诸如所述外围组件接口总线或工业标准架构总线)间的通讯。再者，所述连结系如图所示，可以在输入 / 输出装置之间使用菊链式架构以非耦合性方式而运作。应所述注意的是从其中一个处理节点至另一个所传送的封包可以透过一个或多个的中介节点而传递。例如，由处理节点 612A 传送至处理节点 612D 的封包可以透过如同于图 7 所示的处理节点 612B 或处理节点 612C 而传送。任何适当的递送算法均可使用。计算机系统 400 的其它实施例可以包含比在图 7 中所示的实施例更多或更少的处理节点。

通常，所述封包可以以一个或多个的位次数在节点之间之线路 624 上传送。位次数可以是在所述对应的时脉线路上的时脉讯号的上升或下降缘。所述封包可以包含用于启始处理的指令封包、用于维持缓存耦合性的探测封包及来自于响应探测及指令的响应封包。

除了内存控制器及接口逻辑之外，处理节点 612A 至 612D 可以包含一个或多个的处理器。广泛地说，处理节点包含至少一个处理器并且可以选择性地包含用于与内存及其它所需逻辑通讯的内存控制器。尤其，每一个处理器节点 612A 至 612D 可以包含一个或多个的处理器 100 的复制。每一个处理节点 612 可以包含处理器 100，如同上文所描述，所述处理器 100 包含调度单元 104、跟踪缓存 160、退返队列 102、重启逻辑 190 及 / 或跟踪缓存产生器 170。外部接口单元 18 可以包含在所述节点内部的接口逻辑 618，以及所述内存控制器 616。

内存 614A 至 614D 可以包含任何适当的内存装置。例如，内存 614A 至 614D 可以包含一个或多个的 RAMBUS 动态随机存取内存、同步动态随机存取内存、静态随机存取内存等等。计算机系统 400 的地址空

间在内存 614A 至 614D 间系予分隔开。每个处理节点 612A 至 612D 可以包含用以决定那一个地址为对映至那一个内存 614A 至 614D 的内存对映, 及继而决定特定地址的内存请求应递送至所述处理节点 612A 至 612D 的何者。在一实施例中, 对于地址在计算机系统 400 的内部的耦合点为内存控制器 616A 至 616D 连结至对应于所述地址的内存存储字节处。易言的, 所述内存控制器 616A 至 616D 负责确保存取所述对应内存 614A 至 614D 的每一个内存以缓存耦合方式而产生。内存控制器 616A 至 616D 可以包含用于与内存 614A 至 614D 接口接合的控制电路。此外, 内存控制器 616A 至 616D 可以包含用于等待内存请求的请求队列。

接口逻辑 618A 至 618L 可以包含用于从所述连结接收封包及用于在连结时缓冲所传送的封包的各种缓冲区。计算机系统 400 可以使用用于传送封包的任何适当的流程控制机制。例如, 在一实施例中, 每个接口逻辑 618 存储在所述接口逻辑所连接的连结的另一端处的所述接收器内部的每个形式的缓冲区的数目的计数。所述接口逻辑不传送封包, 除非所述接收接口逻辑具有闲置的缓冲区以存储所述封包。当接收的缓冲区藉由向前传递封包而闲置时, 所述接收接口逻辑传送讯息至所述发出接口逻辑以显示所述缓冲区已经是闲置的。此类的机制可以称为“附加型(coupon-based)”系统。

输入 / 输出装置 620A 至 620B 可以是任何适当的输入 / 输出装置。例如, 输入 / 输出装置 620A 至 620B 可以包含用于与另一个计算机系统通讯的装置, 所述装置可以与所述另一个计算机系统连结(例如网络适配卡或调制解调器)。再者, 输入 / 输出装置 620A 至 620B 可以包含视讯加速器、音讯卡、硬式或软式磁盘驱动器或驱动控制器、小型计算机系统接口(Small computer system Interface, SCSI)转接器及电话卡、声卡及各种资料撷取卡, 诸如通用接口总线或字段总线适配卡。应所述注意的是所述名词“输入 / 输出装置”及所述名词“外围装置”在此是意指同义的。

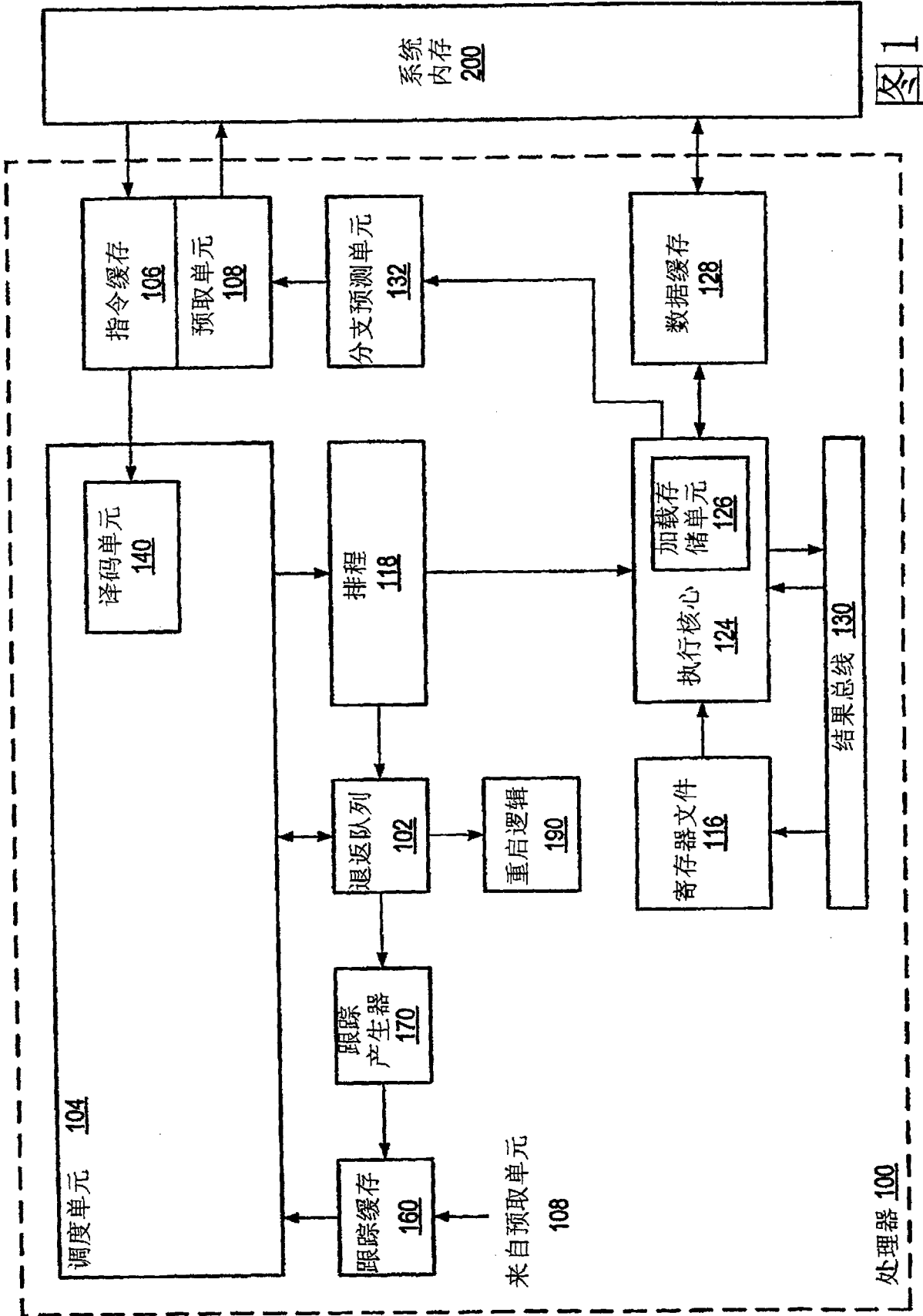
如同在此所使用的, 所述名词“时脉周期”或“周期”意指其中所述指令处理管线的各种行程完成本身任务的时间的间隔。指令及计算的数值是藉由内存组件(诸如寄存器或数组)依据定义所述时脉周期的时脉

讯号而捕捉。例如，内存组件依据所述时脉讯号的上升或下降缘可以捕捉数值。

对于熟习此项技艺的人士在完全了解上文的揭示时，各种变化及修正将变得明显。所述下文的权利要求书应所述解释为涵括所有此类的变化及修正。

工业应用

本发明通常可应用于处理器的领域中。



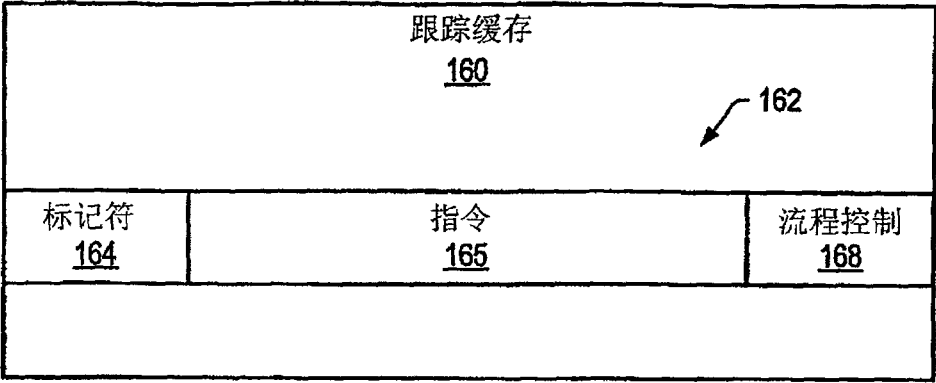


图2

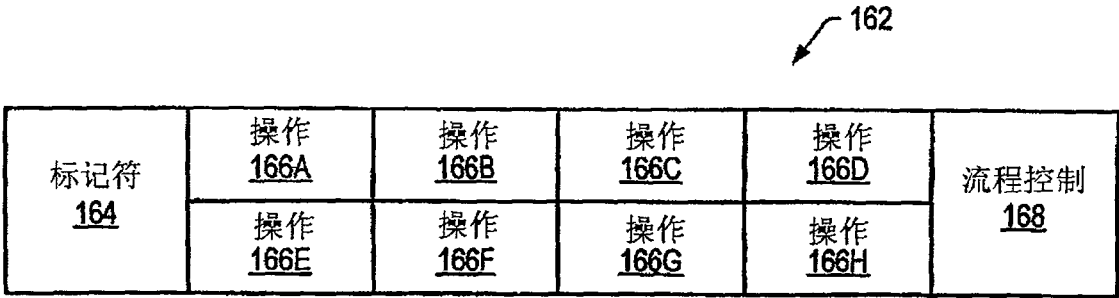


图3A

存在	编码 266
无条件存在	11
后来的第1分支	10
后来的第2分支	01
无条件停滞	00

图3B

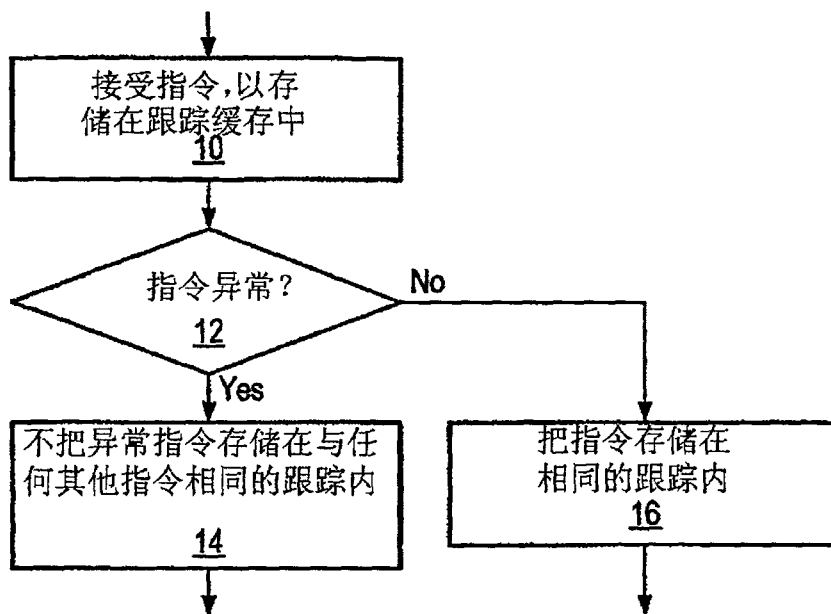


图4

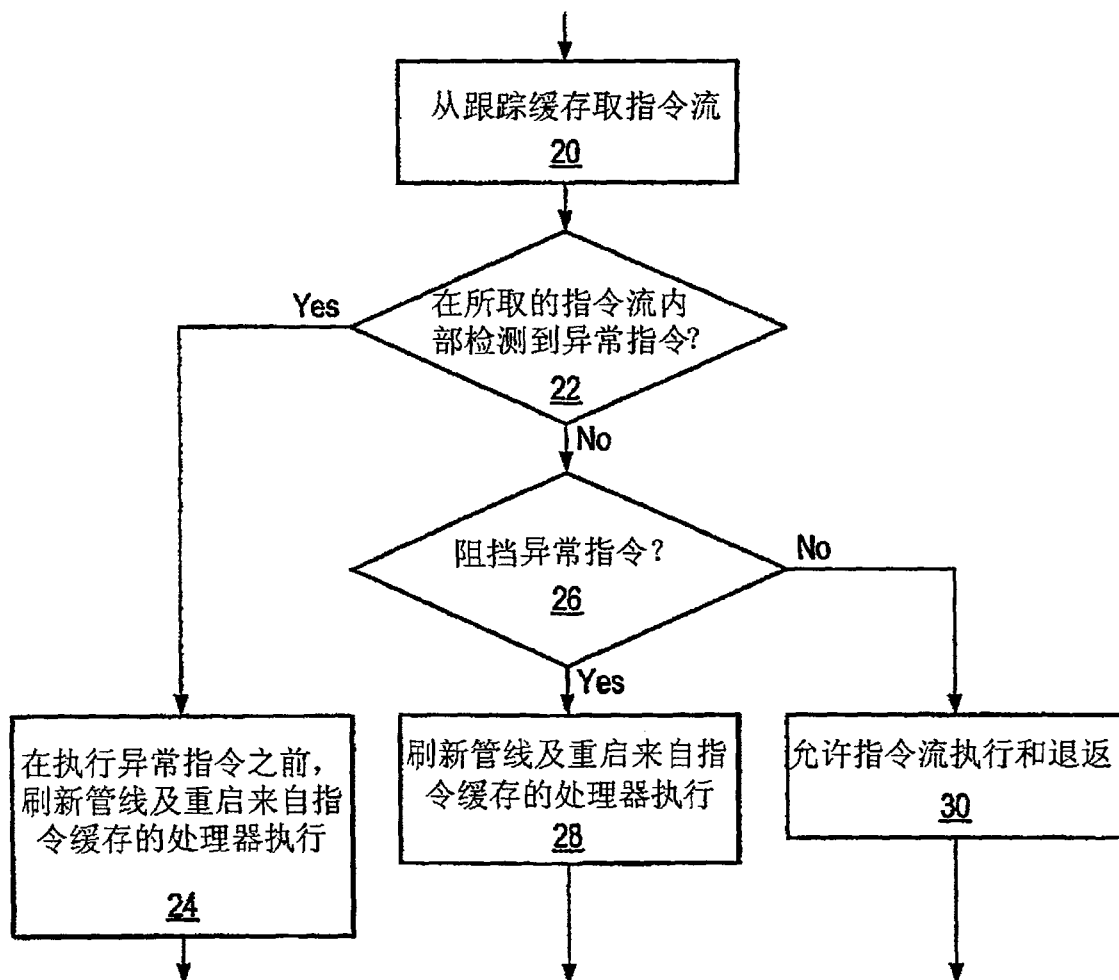


图5

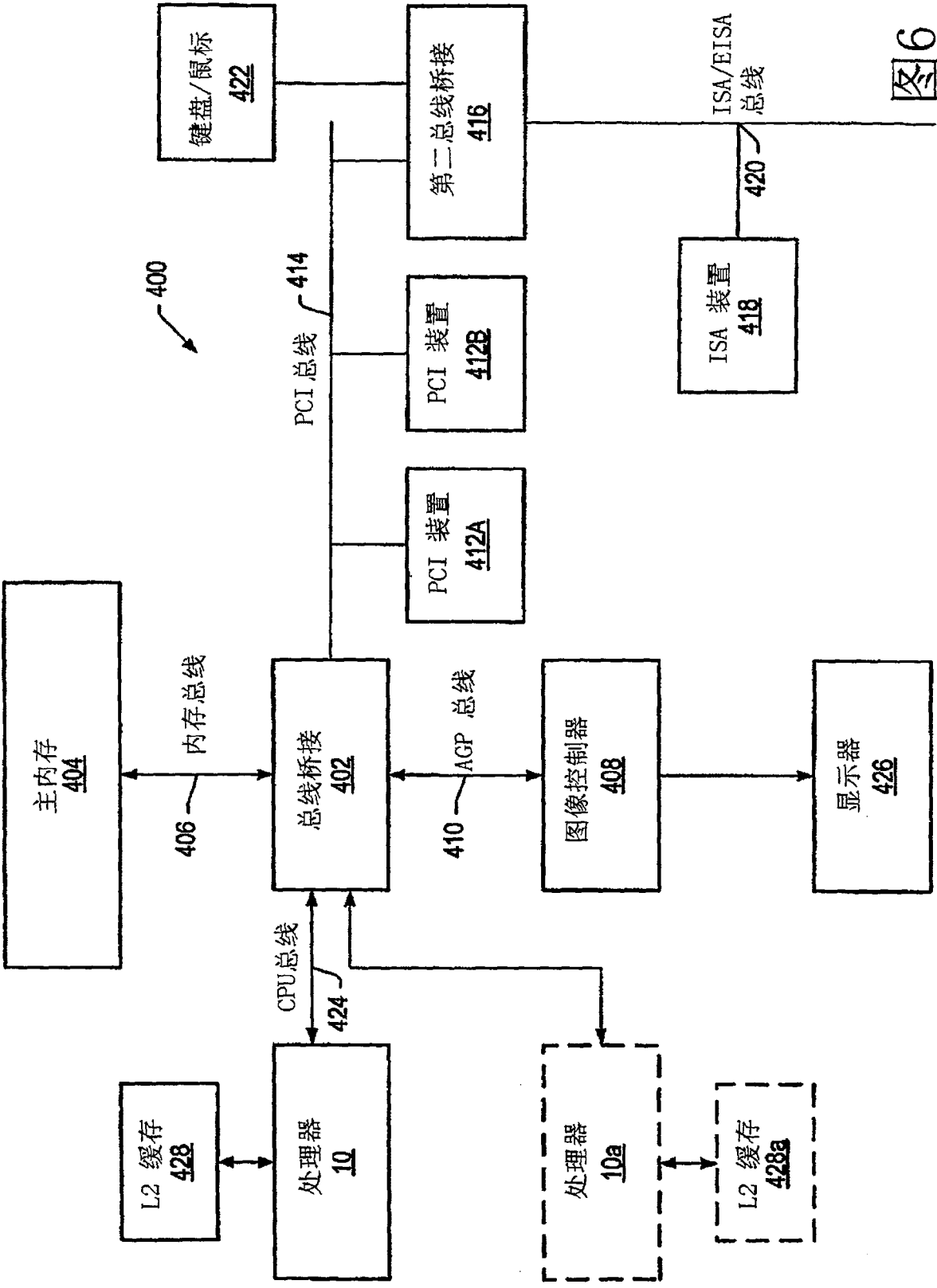


图6

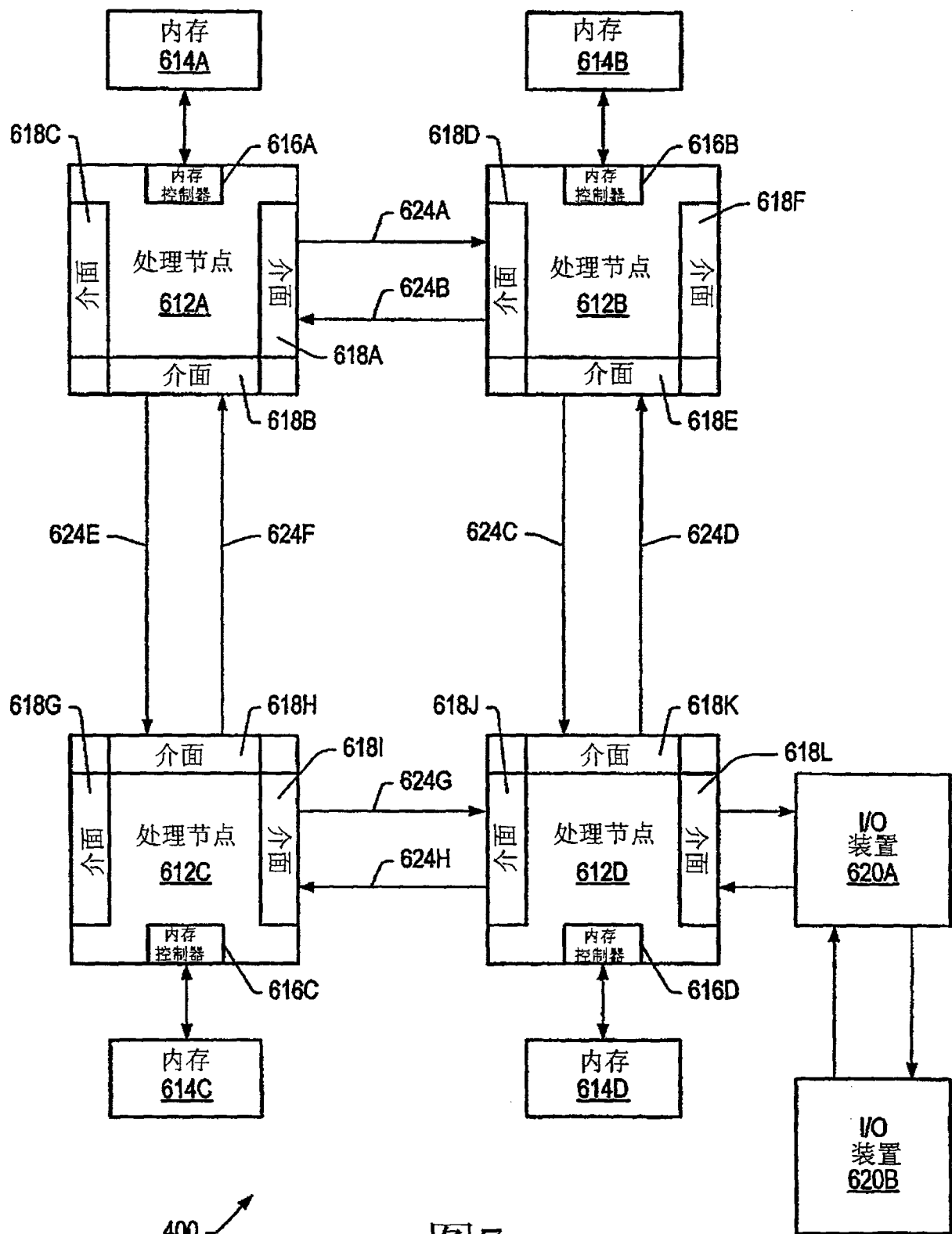


图7