

(12) 发明专利

(10) 授权公告号 CN 1804792 B

(45) 授权公告日 2011. 01. 26

(21) 申请号 200510119155. 8

(22) 申请日 2005. 12. 15

(30) 优先权数据

11/015874 2004. 12. 16 US

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 R·拉瓦 S·斯里尼瓦桑

H·阿卡里 A·甘地

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 陈景峻

(51) Int. Cl.

G06F 9/38(2006. 01)

审查员 刘慧

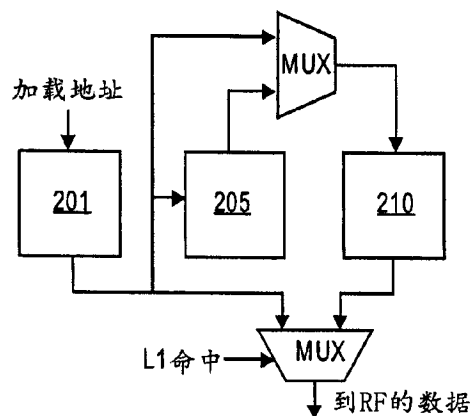
权利要求书 2 页 说明书 6 页 附图 7 页

(54) 发明名称

在长等待时间指令执行期间允许存储转发的方法和系统

(57) 摘要

一种技术, 允许在高等待时间指令处理期间满足独立加载。本发明的实施例涉及这样的技术: 其中在处理高等待时间指令过程中, 当满足独立加载指令时, 存储结构用于按照程序顺序保存存储操作。在处理高等待时间指令之后, 能够按照程序顺序恢复存储操作而无需查找存储结构。



1. 一种在长等待时间指令执行期间允许存储转发的方法,包括:
发出高等等待时间指令;
将按照程序顺序在所述高等等待时间指令之后的多个存储指令存储在先入先出队列中;
在所述高等等待时间指令检索数据时,通过独立于所述高等等待时间指令的加载指令来从由独立于所述高等等待时间指令的存储指令存储的数据中读取要被加载的加载数据。
2. 根据权利要求 1 所述的方法,进一步包括在所述高等等待时间指令检索数据时将所述加载数据暂时存储在数据高速缓存或 1 级 L1 存储队列中,使得所述加载指令能够读取所述暂时存储的加载数据。
3. 根据权利要求 2 所述的方法,进一步包括在所述高等等待时间指令检索数据时将从属存储指令存储在等待缓冲器中。
4. 根据权利要求 3 所述的方法,进一步包括在所述数据高速缓存中按照程序顺序重新装配与所述等待缓冲器中存储的所述从属存储指令相关联的数据以及与所述先入先出队列中存储的存储指令相关联的数据。
5. 根据权利要求 1 所述的方法,其中所述先入先出队列按照程序顺序存储独立和从属存储指令,使得可将所述队列的内容按照程序顺序存储在数据高速缓存中。
6. 根据权利要求 5 所述的方法,其中所述高等等待时间指令从动态随机存取存储器中检索数据。
7. 根据权利要求 2 所述的方法,其中如果所述存储的数据被存储到与之前存储的有效数据相对应的所述数据高速缓存的区域,那么所述之前存储的有效数据被暂时存储在另一个存储位置,并且在所述高等等待时间指令完成执行后所述有效数据被恢复到所述数据高速缓存的区域。
8. 根据权利要求 2 所述的方法,其中要被所述加载指令加载的所述数据是存储在所述 1 级 L1 存储队列中的与所述指令相对应的数据。
9. 根据权利要求 8 所述的方法,其中具有用于所述加载指令的加载数据的所述先入先出队列中存储的所述指令独立于所述高等等待时间指令。
10. 一种在长等待时间指令执行期间允许存储转发的系统,包括:
存储器,用于存储第一数据;
处理器,用于执行高等等待时间指令以从所述存储器检索所述第一数据;
队列,用于在所述高等等待时间指令检索所述第一数据时,按照程序顺序存储多个存储指令;
数据高速缓存,用于暂时包括与所述多个存储指令中的至少一个相对应的存储数据,所述存储数据为独立于所述高等等待时间指令的加载指令提供加载数据;
组相关加载缓冲器,用于存储与所述加载指令相对应的、将与所述存储指令的目标地址进行比较的地址。
11. 根据权利要求 10 所述的系统,其中所述数据高速缓存在所述高等等待时间指令完成执行之后并在已经从所述队列中读取所述多个存储指令之后按照程序顺序存储与所述存储指令相对应的多个数据。
12. 根据权利要求 11 所述的系统,进一步包括弱检测过滤器,以允许所述加载操作从

存储在所述队列中的数据中检索加载数据,而不是从所述数据高速缓存中读取所述存储的数据。

13. 根据权利要求 12 所述的系统,进一步包括等待缓冲器,用于存储依赖于所述高等等待时间指令的指令。

14. 根据权利要求 13 所述的系统,其中存储第一数据的存储器包括动态随机存取存储器。

15. 根据权利要求 14 所述的系统,其中所述队列是先入先出队列。

16. 根据权利要求 15 所述的系统,其中所述弱检测过滤器包括按存储器地址的散列函数索引的直接映射非标记计数器数组。

17. 根据权利要求 16 所述的系统,其中如果存储指令存储在所述队列之中,则使所述弱检测过滤器中的相应计数器递增。

18. 根据权利要求 17 所述的系统,其中所述队列、所述数据高速缓存、所述等待缓冲器以及所述弱检测过滤器都包含在微处理器内。

在长等待时间指令执行期间允许存储转发的方法和系统

技术领域

[0001] 本发明的实施例涉及微处理器和微处理器系统。更为具体地,本发明的实施例涉及能够存储要转发的数据,以便加载指令必须搜索队列以查找要转发的存储的技术。

[0002] 背景技术

[0003] 当执行加载和存储指令时,通常现有技术微处理器依赖于包含内容可编址存储器(CAM)逻辑的可查找队列,以增强存储器操作间的排序,并且在高等时间指令从存储器存取数据时(“未决”)用于将与存储指令相对应的数据转发到加载指令。如果相应数据没有出现在相对较快的存储结构中、例如高速缓存,那么必须借助于具有相对较慢的存取时间的存储器结构如动态随机存取存储器(DRAM)的指令,可能产生高延迟指令。在特定存储结构中缺少所需数据通常称为“未命中”,而存储结构中存在所述数据通常称为“命中”。

[0004] 图1说明现有技术处理器结构体系,包括用于执行指令的逻辑,所述指令独立于高等时间指令(high-latency instruction)。图1的现有技术结构体系能够不断地执行指令而不用停止处理器,包括独立于长等待时间指令的指令,诸如从相对较慢的存储资源(例如DRAM)存取数据的加载。特别地,由指令解码器解码的指令以及由分配和寄存器换名单元分配的寄存器都作为微指令(uop)存储在微指令队列中,从微指令队列中调度这些微指令用于由功能单元执行并将这些微指令发送至寄存器文件。

[0005] 图1的现有技术结构体系允许独立于未命中的指令通过强制长等待时间指令来使用寄存器文件和调度程序资源,并且那些指令依赖于长等待时间指令来缓和调度和寄存器文件资源,直到未命中能够能得到服务。这使得独立于未命中的指令执行和完成,而不被长等待时间指令或他们的从属指令阻塞。

[0006] 在图1中,依赖于长等待时间指令的指令暂时存储在等待缓冲器中,而独立指令在长等待时间指令的未决期间得到服务。然而,为了确保正确的存储排序,必须在长等待时间指令的未决期间存储所有并行进行(“在处理”)中的存储指令,通常需要大的存储队列(例如,L1和L2存储队列)。这些存储队列能够随指令处理的增多而增大。

[0007] 而且,为了查找这些存储队列,诸如CAM逻辑之类的额外逻辑是必需的。特别是,加载操作(查找具有数据的相应存储操作以满足加载操作)通常利用随队列大小而增长的CAM逻辑来查找相对较大的存储队列。

[0008] 查找具有CAM逻辑的大存储队列能够潜在增加周期时间或增加存取存储队列所花费的周期数量。而且,在处理长等待时间指令过程中,当正在处理的指令数量增加的时候,使用可查找存储队列将存储数据转发到合适的加载指令会变得日益困难,例如服务未命中的加载。而且,诸如CAM逻辑之类的、通常与可查找存储队列相关联的查找逻辑为了在未决长等待时间操作期间满足独立的加载操作可能需要额外的能量、芯片空间和处理周期。

发明内容

[0009] 一种装置,包括第一存储设备,用于在高等时间操作从第二存储设备存取数据

期间记录多个存储操作的程序顺序。

[0010] 一种方法,包括:发出高等待时间指令;将按照程序顺序在高等待时间指令之后的多个存储指令存储在先入先出队列中;在高等待时间指令检索数据时,利用独立于高等待时间指令的存储指令的数据来满足独立于高等待时间指令的加载指令。

[0011] 一种系统,包括:存储器,用于存储第一数据;处理器,用于执行高等待时间指令以检索第一数据;队列,用于在高等待时间操作检索第一数据时,按照程序顺序存储多个存储指令;高速缓存,用于暂时包括与多个存储指令中的至少一个相对应的存储数据,存储数据满足独立于高等待时间操作的加载指令;组相关加载缓冲器,用于存储与加载指令相对应的存储地址,存储地址要与存储指令的目标地址相比较。

[0012] 一种机器可读介质,具有存储在其上的一组指令,如果由机器执行指令,则使机器执行一种方法,该方法包括:响应需要执行多于最小时间的存储器存取指令,将多个存储操作按照程序顺序存储在先入先出(FIFO)队列中;响应存储器存取指令的完成,从FIFO队列中检索多个存储操作并将它们的数据按照程序顺序存储在数据高速缓存中而不必查找FIFO队列。

附图说明

[0013] 通过举例而非限定的方式说明本发明的实施例,附图中相同的标记表示相似的元件,其中:

[0014] 图1示出微处理器内的现有技术体系结构,用于在未决长等待时间存储器存取操作(诸如加载操作)期间满足独立操作。

[0015] 图2示出根据本发明一个实施例的体系结构,使得满足独立操作而不使用可查找存储结构。

[0016] 图3是示出可以包含在本发明一个实施例中的操作的流程图。

[0017] 图4示出根据一个实施例的体系结构,包括弱检测过滤器(loose-checkfilter),使得满足独立操作而不使用可查找存储结构。

[0018] 图5是共享总线计算机系统,在其中可以使用本发明的一个实施例。

[0019] 图6是点对点计算机系统,在其中可以使用本发明的一个实施例。

[0020] 图7示出根据一个实施例的体系结构,使用加载缓冲器来存储与匹配存储地址相对应的加载地址。

[0021] 图8示出根据一个实施例的图7的加载缓冲器的条目。

具体实施方式

[0022] 本发明的实施例涉及向独立于高等待时间指令(high-latency instruction)(诸如存取DRAM来检索加载数据的加载指令)的指令提供服务的技术。更为具体地,本发明的实施例涉及在高等待时间操作未决时,将存储数据匹配于加载操作,而不使用可查找存储设备的技术。

[0023] 本发明的至少一个实施例替换了具有相关内容可寻址存储器(CAM)逻辑的第二级("L2")队列而支持先入先出(FIFO)队列,所述FIFO队列在特定时间正执行的一个或多个长等待时间操作的影响下保持存储操作(从属的和独立的)。FIFO队列很可能使用更

少的能量,因为 FIFO 队列不需要像一般的 CAM 结构那样的查找以及相关的查找逻辑。而且在至少一个实施例中,独立加载从 L1 数据高速缓存转发数据以有助于保持性能。

[0024] 图 2 示例了本发明的一个实施例,其中 FIFO 队列用于由于高速缓存未命中而在一个或多个操作正存取相对较慢存取存储器(例如动态随机存取存储器(DRAM))时进行存储。更为具体地,图 2 示例了第一级(“L1”)存储队列 201,其中存储了各种指令或微操作。如果诸如指令或微操作之类的操作不能从相对较快存取存储器(例如数据高速缓存)中检索到所需要的数据,从而创建高速缓存未命中条件,那么可以借助于其他存储器如 DRAM 或更高级别并通常更慢的高速缓存来检索数据。下文中对相对较慢存取存储资源(例如 DRAM)进行存取的操作(例如命令或微操作)称为“高等待时间”操作、指令或微操作。在高等待时间操作试图检索另一源中的存储器期间,独立于高等待时间操作的操作(下文称为“独立操作”)不应该由高等待时间操作或依赖于高等待时间操作的操作(下文称为“从属操作”)选通,但应该允许独立操作完成,因为它们不依赖于高等待时间操作。

[0025] 相应地,图 2 示出以程序顺序存储在长等待时间操作之后发生的操作的再存储逻辑(SRL)(store redo log) 205。不象现有技术,SRL 不包含 CAM 逻辑或者查找 SRL 所需的任何其他逻辑,而是 SRL 按照独立和从属存储操作在程序中出现的顺序对它们进行存储,这样在需要时可按程序顺序将它们读出。在一个实施例中,SRL 是 FIFO 队列。而在其他实施例中,SRL 可以是为了从存储结构中检索出指令或微操作而不必对其查找的其他存储结构。

[0026] 在本发明的一个实施例中,通过将所需的独立存储数据存储在 L1 存储队列或存储在 L1 数据高速缓存 210 中,在满足高等待时间操作之前,可以通过 L1 存储队列 201 或者 L1 数据高速缓存 210 中的独立存储操作来满足独立加载操作。数据高速缓存能够作为要由独立加载操作使用的数据的暂时存储位置,并且取决于程序顺序,在长等待时间操作结束之后,可以由对应于长等待时间操作或从属操作的数据替换。而且,之前存储在数据高速缓存中的位置上的由独立存储操作写入的任何数据(“脏块”)都能够被存储至其他存储结构中,例如 L2 高速缓存,并且在长等待时间操作结束之后返回至数据高速缓存。

[0027] 图 2 也示例了等待缓冲器,在高等待时间操作仍未决时存储从属操作。在一些实施例中等待缓冲器可以是 FIFO。然而在其他实施例中,等待缓冲器可以是其他类型的存储结构。在满足了高等待时间操作之后,在数据高速缓存中以程序顺序重新装配从属和独立数据(包括已经写入数据高速缓存中以满足加载操作的数据)。在本发明的一个实施例中,通过将 SRL 的数据按照程序顺序写入并在数据高速缓存中重新装配数据。在将与高等待时间操作对应的数据写入数据高速缓存中的适当位置之后,存储在 SRL 中的存储操作的数据可以被顺序地从 SRL 中读出并存储至数据高速缓存中的适当位置,而不必搜索 SRL 以查找适当的数据。

[0028] 因为与独立和从属操作对应的所有数据都能从 SRL 中顺序地读出,所以相对于使用诸如 L2 存储队列的存储结构来存储独立和从属指令或微操作的现有技术而言,能够更快检索到数据并且消耗更少的能量。而且,SRL 可以小于现有技术的结构,因为 SRL 不包含诸如 CAM 逻辑的查找逻辑。

[0029] 图 3 是示出可以在本发明一个实施例中使用以实现实施例(如图 2 实施例)的一系列操作的流程图。在操作 301,遇到高等待时间指令或者微操作,使得指令借助于相对较

高等待时间存储器（例如 DRAM）来检索指令或微操作需要的数据。当高等待时间指令或微操作存取它需要的数据（即指令或者微操作“未决”）时，在操作 305，依赖于高等待时间指令的指令或微操作都存储在等待缓冲器中，而依赖于和独立于高等待时间指令或微操作的存储指令或微操作存储于 SRL 中。

[0030] 在操作 310，独立于高等待时间指令的指令或微操作可以将适当数据暂时地写入数据高速缓存中，独立指令或者微操作可以从该数据高速缓存读取数据。如果在操作 315 与独立指令相对应的数据被写入脏块中，那么在操作 320，脏块中的数据被暂时存储至其他存储器中，例如 L2 高速缓存。在操作 325，在高等待时间指令已经检索到它的数据之后，可以通过将来自等待缓冲器中存储的指令或者微操作的数据拷贝到 SRL 中其适当的位置而按照程序顺序重新装配独立数据和从属数据，接着将 SRL 中存储的存储指令或者微操作的数据写入数据高速缓存中的相应位置，使得以程序顺序存储数据。

[0031] 图 4 示例了本发明的一个实施例，其中使用弱检测过滤器 (LCF) 403 来识别 SRL 405 是否包含具有满足特定加载操作的数据的指令或者微操作。LCF 的输出连接到多路复用器 407，多路复用器 407 能够直接从 SRL 中选择数据而不必如图 3 所说明的实施例那样首先将数据拷贝到数据高速缓存 410。如图 3 的实施例，数据可以经由多路复用器 413 从 L1 队列 401 或者数据高速缓存中被选择并被写入寄存器文件（未示出）。

[0032] 在本发明的一个实施例中，LCF 是按存储地址的散列函数索引的、直接映射非标记计数器数组。将存储操作插入到 SRL 中使相应的 LCF 计数器递增，而从 SRL 中删除存储操作使 LCF 计数器递减。非零 LCF 计数器值意味着 SRL 中可能匹配的存储操作，而零 LCF 计数值保证没有匹配的存储操作。通过允许加载操作通过 LCF 匹配 SRL 中的数据，加载操作可以只停止在与非零值的 LCF 匹配上。

[0033] LCF 可以导致一些加载操作没有必要地停止，但是为了减少这种停止情况，可以在 SRL 中使用索引的转发。因为许多停止的加载操作是最近获取的在高等待时间操作之后出现的加载操作，但是在 SRL 中的所有存储操作都已经将其数据存储在数据高速缓存中之前，SRL 中的转发存储操作通常是最后插入到 SRL 中的匹配存储操作。因此，在一个实施例中，LCF 被扩展以存储与相应 LCF 计数器相关联的最后插入的存储操作的 SRL 索引。在该实施例中，需要对应于 SRL 中的存储操作的存储数据的到达的加载操作能够快速地在 SRL 中定位最后可能匹配的存储数据条目。加载操作能够读取该条目并且执行完全的地址和年限检测，而不需要经由 CAM 逻辑或者一些其他逻辑查找 SRL。而且，在该实施例中整个 SRL 只需要一个地址比较器。

[0034] 在一些情况下，加载可以被预测为独立于高等待时间指令，但是可以最终依赖于从属存储指令。例如，如果高等待时间指令是加载指令，那么不依赖于高等待时间加载指令的后来的加载指令可能依赖于从属存储指令。在这种情况下，如果存在按照程序顺序出现在符合的加载指令之前的、比符合加载的数据具有符合加载指令的更新数据的从属存储操作，那么由独立存储指令的数据满足的加载指令可能会由错误的数据来满足。

[0035] 图 7 示例了本发明的一个实施例，具有加载缓冲器来存储加载指令的目标地址，所述加载指令的目标地址可以与正在处理中的并将被存储到 SRL705 或队列 701 中的存储指令的目标地址相比较，以确定符合的加载指令是否真正具有最新的存储数据。具体而言，加载缓冲器 703 用于存储与加载操作相对应的目标加载地址，并且能够将目标加载地

址与发送到 SRL 或者队列的存储指令的目标存储地址相比较,有效地“监听”正在处理中的存储目标地址。

[0036] 将加载缓冲器中存储的目标地址与发送到 SRL 或者队列的存储指令的目标地址相比较,并且按照程序顺序在存储指令之后、共享相同目标地址的加载缓冲器内的匹配条目指示加载指令从数据高速缓存中检索的数据不能使它得到满足。在一个实施例中,加载缓冲器是组相关缓冲器,而在其他实施例中也可以使用其他存储结构。

[0037] 如果在 SRL 内遇到这样的存储:与加载操作从数据高速缓存中检索的存储数据不对应,并且按照程序顺序在与从数据高速缓存检索的数据对应的指令之后而按照程序顺序在加载操作之前,那么可以使用错误预测恢复机制来检索最新存储数据。在一个实施例中,SRL 中最新存储数据可能使加载指令得到满足,通过将指令的处理管线冲刷回执行顺序中由加载缓冲器内存储在条目中的检验点指示的特定点,从而将加载指令与最新存储数据对应。

[0038] 图 8 示出根据一个实施例的加载缓冲器的条目,其中每个加载缓冲器条目都包含物理地址标记 801,物理地址标记 801 与 SRL 中的存储操作的相应地址字段相比较。每个条目还包括:有效位 805,指示该条目是否包含有效加载地址;标识字段 810,指示在加载操作错误地满足于数据高速缓存中的存储数据之后执行重新开始的位置(“检验点”);以及存储缓冲器标识字段 815,指示按照程序顺序与加载指令最近的前一存储指令的地址;以及存储缓冲器标识字段 817,指示以前满足了加载的存储。

[0039] 在本发明的一个实施例中,在要满足的每个加载操作从数据高速缓存读取数据之前,相对于 SRL 中存储的存储操作检验加载缓冲器条目。在其他实施例中,该检验是在加载操作从数据高速缓存中检索数据之后实现的。由于加载缓冲器是组相关的,而不是全相关的,例如至少一个实施例能够相对较快地将加载地址和 SRL 中的相应存储地址相比较,这样该比较就可以在高等待时间加载操作的存储器存取周期期间进行。

[0040] 图 5 示例了前端总线(FSB)计算机系统,在其中可以使用本发明的一个实施例。处理器 205 从一级(L1)高速缓存 510 和主存储器 515 中存取数据。在本发明的其他实施例中,高速缓存可以是计算机系统存储层次中的二级(L2)高速缓存或者其他存储器。而且,在一些实施例中,图 5 的计算机系统可以既包括 L1 高速缓存又包括 L2 高速缓存,所述 L1 和 L2 高速缓存包括内含的高速缓存层次,其中相干数据在 L1 和 L2 高速缓存之间共享。

[0041] 图 5 的处理器内示例的是本发明的一个实施例 506。在一些实施例中,图 5 的处理器可以是多内核处理器。

[0042] 主存储器可以由各种存储器源实现,例如动态随机存取存储器(DRAM)、硬盘驱动器(HDD)520 或者包含各种存储设备和技术的、经由网络接口 530 远离计算机系统设置的存储器资源。高速缓存可以设置在处理器内或者处理器的邻近处,例如在处理器的本地总线 507 上。而且,高速缓存可以包含相对较快的存储器单元,例如六晶体管(6T)单元或者大约相同或者更快存取速度的其他存储器单元。

[0043] 图 5 的计算机系统可以是总线代理例如微处理器的点对点(PtP)网络,总线代理经由专用于 PtP 网络上每个代理的总线信号进行通信。本发明 506 的至少一个实施例在每个总线代理内或者至少与每个总线代理相关联,这样在总线代理之间能够以迅速的方式便于存储操作。

[0044] 图 6 示例了以点对点 (PtP) 配置设置的计算机系统。特别地,图 6 显示了这样的系统,其中处理器、存储器和输入 / 输出设备由多个点对点接口互连。

[0045] 图 6 的系统也可以包括多个处理器,为了简洁只示出了两个处理器 670、680。处理器 670、680 每个可以包括本地存储控制器集线器 (MCH) 672、682 来连接存储器 62、64。处理器 670、680 可以使用 PtP 接口电路 678、688 经由点对点 (PtP) 接口 650 交换数据。处理器 670、680 均可以使用点对点接口电路 676、694、686、698 经由各个 PtP 接口 652、654 与芯片组 690 交换数据。芯片组 690 也可以经由高性能图形接口 639 与高性能图形电路 638 交换数据。

[0046] 本发明的至少一个实施例可以设置在处理器 670、680 内。然而本发明的其他实施例可以存在于图 6 的系统中的其他电路、逻辑单元、或者设备之中。而且,本发明的其他实施例可以分布在图 6 所示的多个电路、逻辑单元或者设备。

[0047] 本文所描述的本发明的多个实施例可以利用使用互补金属氧化物半导体设备或者“硬件”的电路、或者使用介质中存储的一组指令(当由机器例如处理器执行时,执行与本发明实施例相关联的操作)或者“软件”来实现。或者,本发明的实施例可以使用硬件和软件的组合来实现。

[0048] 虽然已经参照实施例描述了本发明,但是该描述并不意味着限制意义。对于本发明所属领域的技术人员而言显而易见的是,对示出实施例以及其他实施例的各种修改都被认为是落入本发明的精神和范围之内。

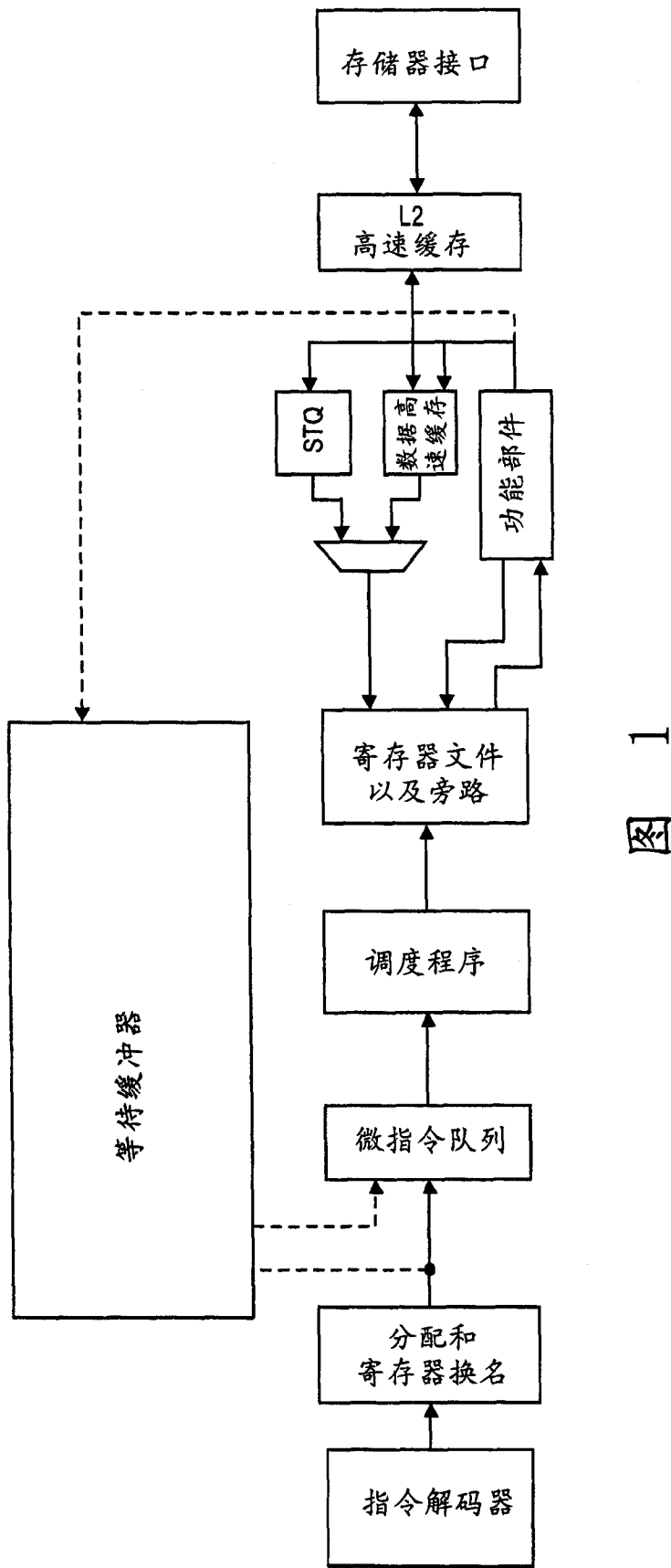


图 1

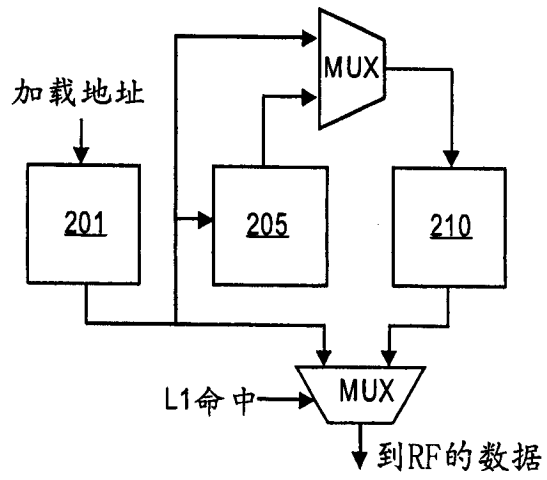


图 2

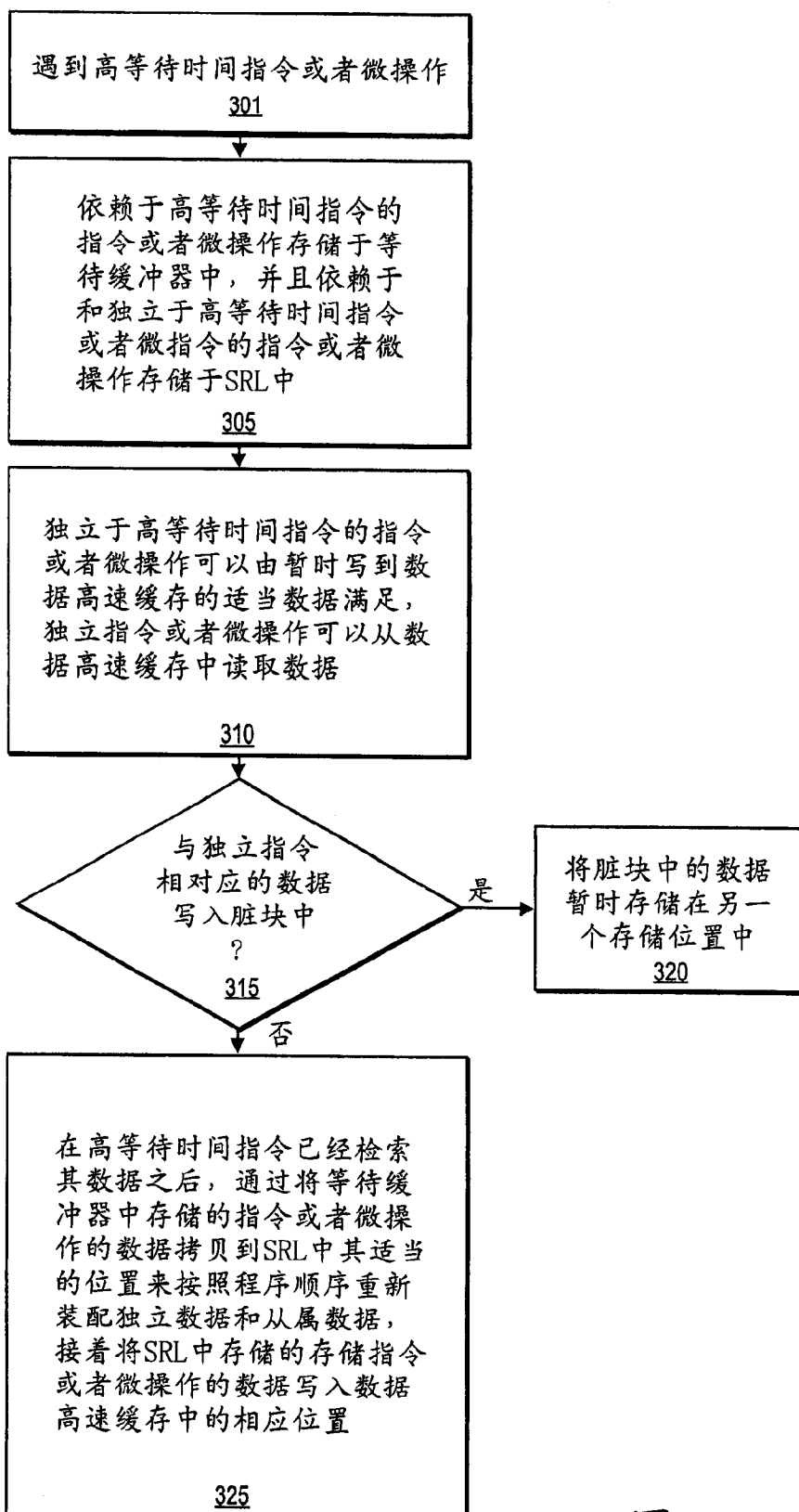


图 3

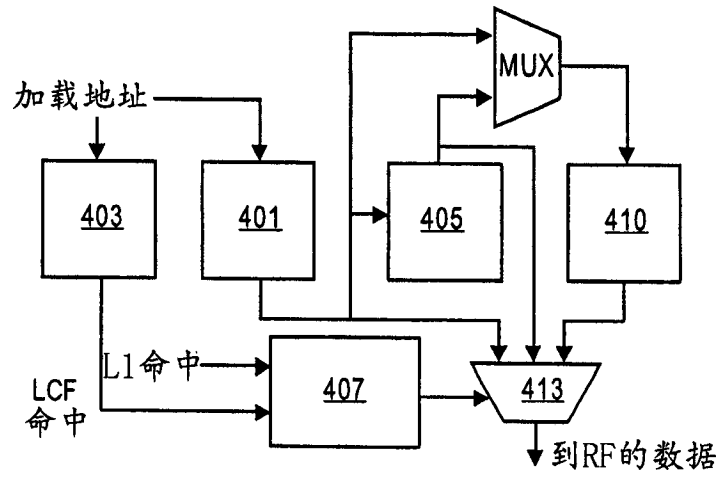


图 4

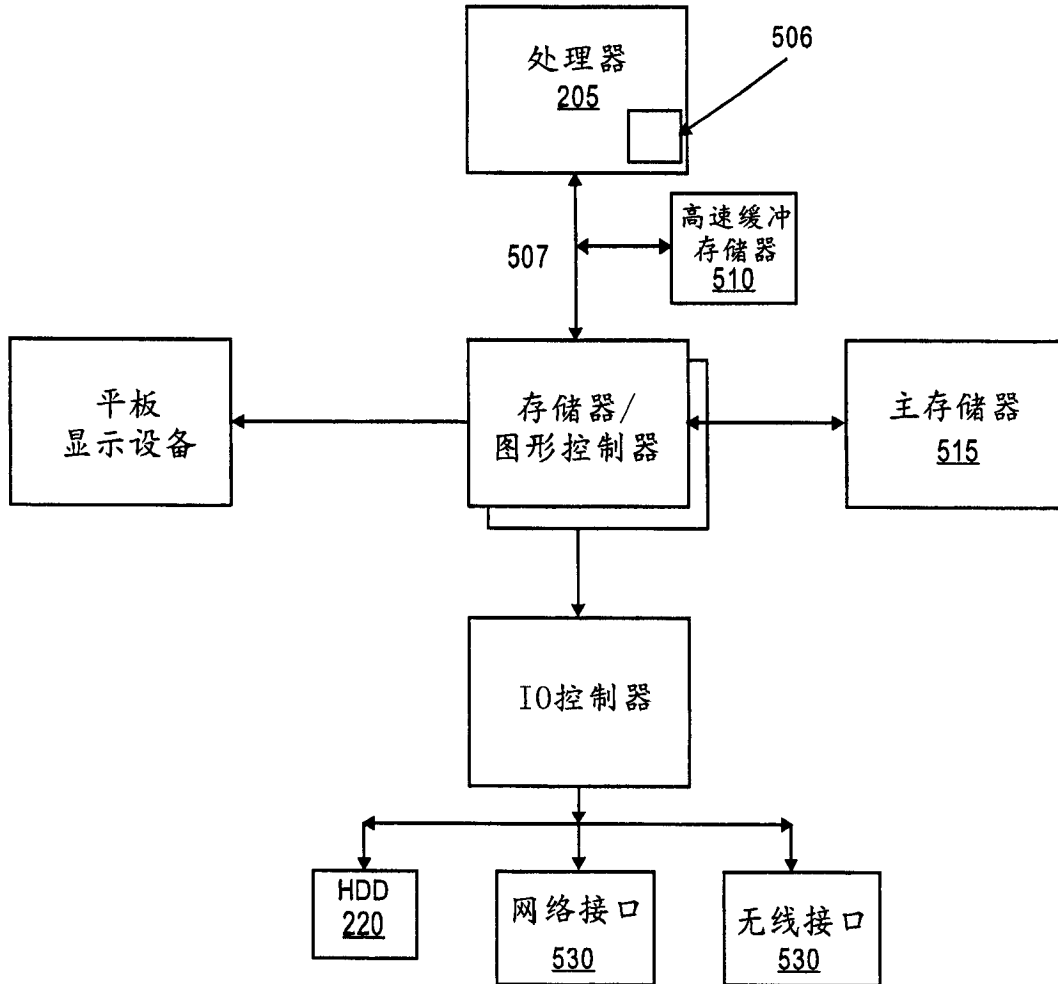


图 5

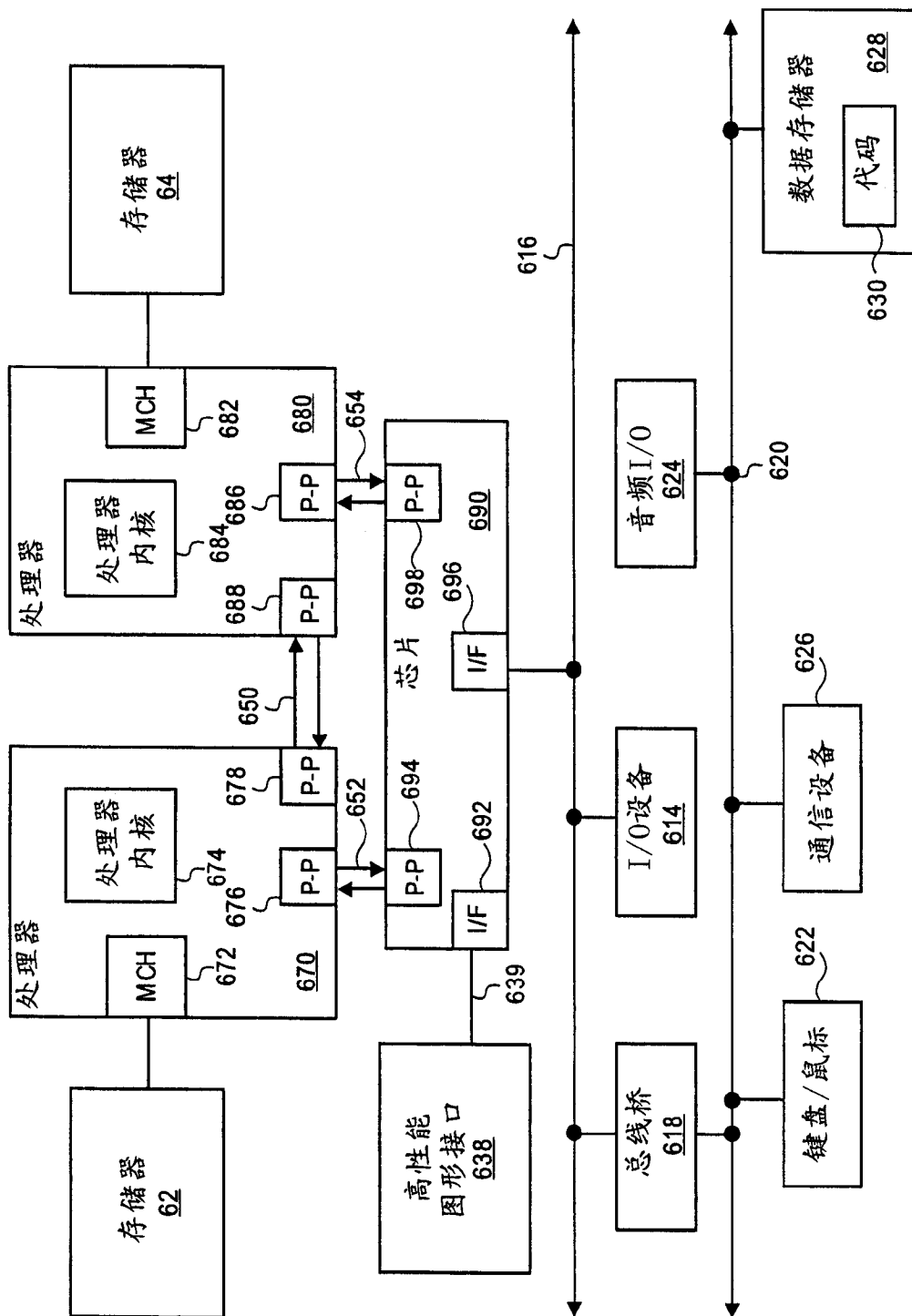


图 6

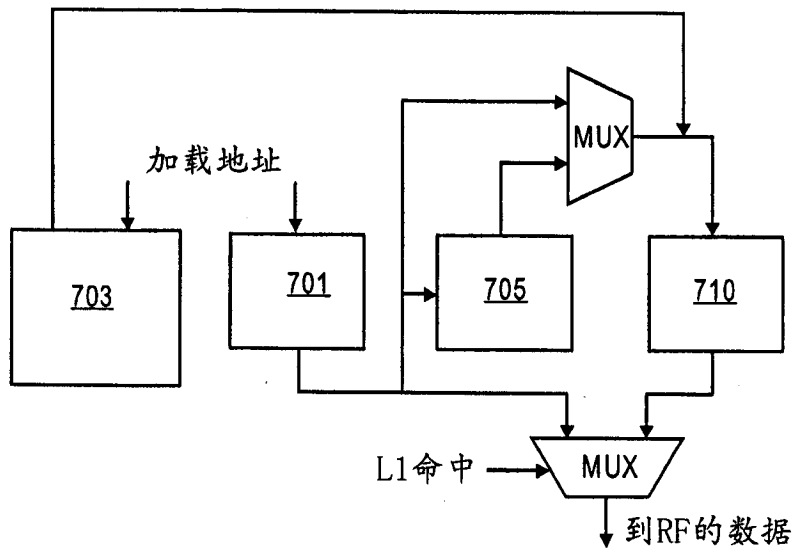


图 7

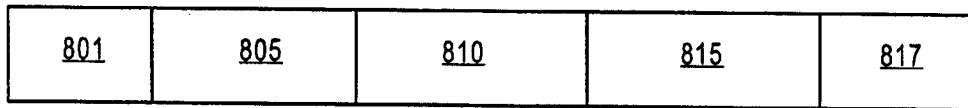


图 8