

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/38 (2006.01)

G06F 9/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610138686.6

[45] 授权公告日 2009 年 11 月 4 日

[11] 授权公告号 CN 100557566C

[22] 申请日 2006.11.10

[21] 申请号 200610138686.6

[73] 专利权人 威盛电子股份有限公司

地址 中国台湾台北县新店市中正路 535 号 8 楼

[72] 发明人 洪伟翔 苏耀群 贾维中 高智国

[56] 参考文献

US6782456B2 2004.8.24

US2006031602A1 2006.2.9

US4716522A 1987.12.29

审查员 段雪莲

[74] 专利代理机构 北京林达刘知识产权代理事务所

代理人 刘新宇

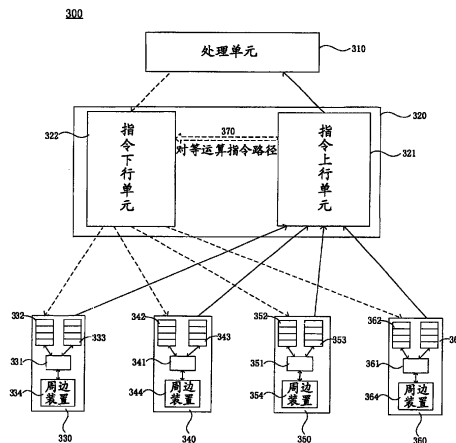
权利要求书 4 页 说明书 10 页 附图 6 页

[54] 发明名称

指令存取及排程的方法及其装置

[57] 摘要

本发明提供一种指令存取及排程的方法及其装置，特别涉及一种指令存取的方法，包括下列步骤：接收至少一指令；判断该指令为一写入指令或非写入指令，其中若该指令为该写入指令则将该指令存储于一元件写入暂存器中，若该指令为该非写入指令，则将该指令存储于一元件非写入暂存器中；判断该指令是否为一对等运算指令，其中若该指令为一对等运算指令则将该指令经由一对等运算指令路径传送至一指令下行单元，若该指令为非对等运算指令则将该指令传送至一处理单元；及接收该对等运算指令路径及该处理单元传送过来的指令并分别下行该写入指令及该非写入指令。本发明可有效地避免系统死机的问题，而使得整个系统能够顺利地运作。



1. 一种指令存取的方法，其特征在于，用以存取至少一指令，其中该指令可为一写入指令或者一非写入指令，所述指令存取的方法包含：

若该指令为该写入指令，则储存该指令至一元件写入暂存器；

若该指令为该非写入指令，则储存该指令至一元件非写入暂存器；以及

判断并排程该写入指令以及该非写入指令的优先序；

其中若该指令为一非对等运算指令，则使得该指令由一处理单元处理；其中若该指令为一对等运算指令，则使得该指令经由一对等运算路径传送至一指令下行单元。

2. 根据权利要求1所述的指令存取的方法，其特征在于，更包含先判断该指令为该对等运算指令或是该非对等运算指令，再分别储存该指令至该元件写入暂存器或是该元件非写入暂存器。

3. 根据权利要求1所述的指令存取的方法，其特征在于，若该指令为一对等运算的写入指令，且该指令之前有一对等运算非写入指令的等待时间超过一预定时间，则使得该指令的优先序超过该对等运算非写入指令，其中利用一第一对等运算路径传送该对等运算非写入指令至该指令下行单元；以及利用一第二对等运算路径传送该对等运算的写入指令至该指令下行单元。

4. 根据权利要求1所述的指令存取的方法，其特征在于，包含该指令下行单元接收并判断由该处理单元以及该对等运算路径传送的该指令的优先序。

5. 一种指令排程的方法，其特征在于，所述指令排程的方法包括：

接收一指令；

其中若该指令为一对等运算的一写入指令，且其中该写入指令之前有对等运算的一非写入指令的等待时间超过一预定时间，

则使得该写入指令的优先序超前该非写入指令。

6. 根据权利要求5所述的指令排程的方法，其特征在于，若该指令为一非对等运算指令，则将该指令传送到一处理单元。

7. 根据权利要求5所述的指令排程的方法，其特征在于，若该指令之前没有该非写入指令，或者该非写入指令的等待时间没有超过该预定时间，则使得该指令经由一对等运算指令路径传送到一指令下行单元。

8. 根据权利要求5所述的指令排程的方法，其特征在于，利用一第一对等运算指令路径传送该非写入指令至一指令下行单元，以及利用一第二对等运算指令路径传送该写入指令至该指令下行单元。

9. 一种快速周边连接装置指令存取装置，其特征在于，所述快速周边连接装置指令存取装置包括：

至少一周边元件，用以发出至少一指令，其中若该指令为一写入指令，则储存该指令于一元件写入暂存器，其中若该指令为一非写入指令，则储存该指令于一元件非写入暂存器；

一芯片，连接至该周边元件用以存取该指令；及

一处理单元，与该芯片连接用以处理该指令。

10. 根据权利要求9所述的快速周边连接装置指令存取装置，其特征在于，每一个该周边元件更包含一基板地址范围译码单元，接收并判断该指令为一对等运算指令或是一非对等运算指令；以及一周边装置，连接至该基板地址范围译码单元，用以发出该指令。

11. 根据权利要求9所述的快速周边连接装置指令存取装置，其特征在于，该芯片包含有：

一指令上行单元，用以分别接收由每一个该元件写入暂存器以及每一个该元件非写入暂存器所上传的该指令；以及

一指令下行单元，接收由该指令上行单元以及该处理单元传递过来的该指令，并传送至对应的该周边元件。

12. 根据权利要求11所述的快速周边连接装置指令存取装置，其特征在于，该指令上行单元包含有：

一上行写入暂存器；

一上行写入指令裁决端口，连接于该元件写入暂存器以及该上行写入暂存器间，用以接收并判断存储于该元件写入暂存器的该指令的优先序，之后将该指令储存于该上行写入暂存器；

一上行非写入暂存器；

一上行非写入指令裁决端口，连接于该元件非写入暂存器以及该上行非写入暂存器间，用以接收并判断储存于该元件非写入暂存器的该指令的优先序，之后将该指令储存于该上行非写入暂存器；以及

一上行排程器，连接至该上行写入暂存器以及该上行非写入暂存器，用以排程储存于该上行写入暂存器以及该上行非写入暂存器的写入指令以及非写入指令。

13. 根据权利要求12所述的快速周边连接装置指令存取装置，其特征在于，若该指令为非对等运算指令，则该上行排程器传送该指令至该处理单元；若该指令为对等运算指令，则该上行排程器经由一第一对等运算指令路径传送该指令至该指令下行单元；其中若该指令为一对等运算的写入指令，且在该指令之前有一对等运算的非写入指令的等待时间超过一预定时间，则该上行排程器使得该对等运算的写入指令的优先序超过该对等运算的非写入指令；其中利用一第二对等运算指令路径传送该对等运算的写入指令，使得该对等运算的写入指令较该对等运算的非写入指令早传送至该指令下行单元。

14. 根据权利要求12所述的快速周边连接装置指令存取装置，

其特征在于，该指令下行单元包含有：

一下行排程器，用以分别接收由该处理单元以及该指令上行单元传送过来的该指令；

一下行写入指令译码器，连接至该下行排程器，接收并译码由该下行排程器传送过来的该写入指令；

一下行写入暂存器，连接于该下行写入指令译码器以及每一个该周边元件间，储存该下行写入指令译码器译码过后的该写入指令，并传送至对应的该周边元件；

一下行非写入指令译码器，连接于该下行排程器，接收并译码由该下行排程器传送过来的该非写入指令；以及

一下行非写入暂存器，连接于该下行非写入指令译码器以及每一个该周边元件间，储存该下行非写入指令译码器译码过后的该非写入指令，并传送至对应的该周边元件。

指令存取及排程的方法及其装置

技术领域

本发明是有关于一种存取及排程指令的方法及其装置，尤指一种存取及排程对等运算指令的方法及其装置。

背景技术

随着处理器及系统性能的不断提高，采用串联、点对点连接架构的高速周边元件连接接口(Peripheral Components Interconnect Express, PCIE)被越来越多的应用在计算机系统中。计算机系统的芯片组通过PCIE总线与各周边元件相连接，使得各周边元件发出的指令通过芯片组传送至中央处理器进行运算。

在PCIE的架构中，芯片组对于上行指令的基板地址范围(onboard address range)及高速周边元件连接接口存储器范围(peer-to-peer address range)并没有进行译码的动作，因此所有的上行指令不管是基板指令(onboard access)或是对等运算指令(peer-to-peer access)均传送至中央处理器进行运算，而中央处理器运算完之后再将指令下行至各个周边元件。

图1为已知计算机系统的指令存取方块图。各周边装置121~124分别送出指令至芯片组10，这些指令分别以一队列(Queue)的形式分别储存在暂存器111~114。裁决端口120决定这些上行指令的优先顺序，亦即这些指令传送给中央处理器140的先后顺序。指令在中央处理器140运算完后，下行至译码器130。译码器130对该下行指令进行位地址译码的动作以决定下行指令所要传送的目标周边元件。

在上述的架构中，不管是基板指令(onboard access)还是对等

运算(Peer-to-Peer, P2P)均需传送至中央处理器140进行运算。但事实上,对等运算指令是各个周边元件151~154间互相动作而不须经由中央处理器140运算。如果对等指令也需要传送至中央处理器140,如此会使得指令的等待时间(latency)过长。而且如果指令为存储器写入指令,中央处理器140会延长将数据从中央处理器140写入存储器的时间,并延长读取下一指令的时间,亦即延迟上行存储器存取指令,导致整体表现会有明显的下降。

发明内容

本发明是提出一种存取及排程指令的方法及其装置,其可有效地避免传统作法中系统死机的问题,而使得整个系统能够顺利地运作。

本发明提出一种指令存取的方法,用以存取至少一指令,其中该指令可为一写入指令或者一非写入指令,该方法包含:若指令为写入指令,则将指令储存至一元件写入暂存,若指令为非写入指令,则将指令储存至一元件非写入暂存器。接着判断并排程写入指令以及非写入指令的优先序,其中若指令为一非对等运算指令,则使得指令由一处理单元处理,若指令为一对等运算指令,则使得指令经由对等运算路径传送至一指令下行单元。

本发明又提出一种指令排程的方法,包括:接收并判断一指令是否为一对等运算指令,若指令为该对等运算指令,则判断指令之前是否有一非写入指令的等待时间超过一预定时间。若非写入指令的等待时间超过该预定时间,且指令为一写入指令,则使得指令的优先序超前非写入指令。

本发明又提出一种指令排程的方法,包括:接收一指令,其中若指令为对等运算的一写入指令,且其中写入指令之前有对等运算的一非写入指令的等待时间超过一预定时间,则使得写入指

令的优先序超前非写入指令。

本发明又提出快速周边连接装置指令存取装置，包括：至少一周边元件，用以发出至少一指令，其中若指令为一写入指令，则将指令储存于一元件写入暂存器，若指令为一非写入指令，则将指令储存于一元件非写入暂存器。一芯片，连接至周边元件用以存取该指令；以及一处理单元，与芯片连接用以处理指令。

本发明可有效地避免系统死机的问题，而使得整个系统能够顺利地运作。

附图说明

图1绘示为计算机系统中芯片组的示意图。

图2绘示为另一计算机系统中芯片组的示意图。

图3绘示为本发明的一个实施例的装置的方块图。

图4绘示为本发明的一个实施例的装置的详细方块图。

图5绘示为本发明的一个实施例的方法的流程图。

图6绘示为本发明的又一实施例的方法的流程图。

具体实施方式

为了让本发明的上述目的、特征和优点能更明显易懂，下文特举一较佳实施例，并配合所附图式，作详细说明如下。

为了解决上述中央处理器140效能的问题，图2在芯片组中设有一对等运算指令路径。当周边元件251~254送出指令至芯片组20时，上行译码器240对所述指令进行地址范围译码的动作以判断指令是否为对等运算指令。若指令为非对等运算指令(例如：基板指令)则传送至中央处理器250进行运算，中央处理器250运算完后再将指令下行至下行裁决端口270；若指令为对等运算指令，则直接将指令经由对等运算指令路径260送至下行裁决端口270。

下行裁决端口270决定由中央处理器250下行的非对等运算指令及由对等运算指令路径260传送的对等运算指令下行的先后顺序并传送至下行译码器280。下行译码器280对下行指令进行地址译码的动作，以决定下行指令所要传送的目标周边元件251~254。

在上述的架构中，由于对等运算指令不再送至中央处理器250进行运算，而是经由对等运算指令路径260直接传送，因此大幅节省中央处理器250所占用的资源。此外，通过对等运算指令路径260的设置缩短了对等运算指令的延迟，整体的效能表现得以提升。

然而，上述的架构在某些情况下依然会发生问题。例如，若周边元件251发出一读取周边元件252上的数据的对等运算读取指令，而周边元件252也发出一将数据写入周边元件253的对等运算写入指令。因为读取指令必须在写入指令完成后才可以被执行。因此会使得读取指令与写入指令互相等待的恶性循环，并使得整个系统死机。

为了解决上述的问题，本发明提供了一种快速周边连接装置指令存取装置，一种指令存取的方法以及一种指令排程的方法，将暂存器区分成写入暂存器以及非写入暂存器，并且若有非写入指令需要等待写入指令从缓冲区清除(flush)(亦即等待写入指令被处理完毕)时，使得写入指令由对等运算指令路径超前非写入指令，以避免造成系统死机。

为了说明本发明指令存取的方法，请参照图3。图3为利用本发明的一快速周边连接装置指令存取装置300。如图3所示，快速周边连接装置指令存取装置300包含有：处理单元310，芯片320，以及多个周边元件，在本实施例当中，假设共有四个周边元件330、340、350以及360。芯片320连接于处理单元310与周边元件330~360之间用以快速存取由周边元件330~360发出的指令。而每一个周边元件330~360分别包含有一产生指令的周边装置334、

344、354、364，一基板地址范围译码单元331、341、351以及361用以接收及译码指令是否为对等运算指令。另外也分别有一个元件写入暂存器332、342、352以及362，以及一个元件非写入暂存器333、343、353以及363用以分别储存写入指令(posted)以及非写入指令(non - posted)。芯片320包括有一个指令上行单元321以及一个指令下行单元322。指令上行单元321分别接收由周边元件330~360所上传的指令，如果指令不是对等运算指令，则上传到处理单元310处理，如果指令是对等运算指令，则利用对等运算指令路径370传送到指令下行单元322处理。

请参考图4，详细的绘示图3中指令上行单元321以及指令下行单元322的内部方块图。

如图4所示，指令上行单元321包含有：一上行写入指令裁决端口3211，一上行非写入指令裁决端口3212，一上行写入暂存器3213，一上行非写入暂存器3214以及一上行排程器3215。指令下行单元322包含有：一下行排程器3221，一下行写入指令译码器3222，一下行非写入指令译码器3223，一下行写入指令暂存器3224，以及一下行非写入指令暂存器3225。

以下举例说明本发明指令存取的方法。当周边装置334~364分别发出指令后，基板地址范围译码单元331~361会分别译码这些指令以判断是否为对等运算指令。之后判断指令为写入(posted)指令或是非写入(non - posted)指令，然后分别储存于元件写入暂存器332~362以及元件非写入暂存器333~363中。随后储存于元件写入暂存器332~362以及元件非写入暂存器333~363中的指令分别被传送到指令上行单元321中。

指令上行单元321中的上行写入指令裁决端口3211以及上行非写入指令裁决端口3212分别决定元件写入暂存器332~362以及元件非写入暂存器333~363中的指令的先后顺序，并分别储存到

上行写入暂存器3213以及上行非写入暂存器3214中。而上行排程器3215将上行写入暂存器3213以及上行非写入暂存器3214中的非对等运算指令的指令上传到处理单元310，由处理单元310处理完后再下行到指令下行单元322。另外上行排程器3215将上行写入暂存器3213以及一上行非写入暂存器3214中的对等运算指令的指令经由对等运算指令路径370传送到指令下行单元322。

其中如果有一个先发出的非写入指令的等待时间超过一预设时间T(亦即有先发出的非写入(non - posted)指令需要等待后发出的写入(posted)指令从暂存区中清除(flush))，则使得后发出的写入指令的优先序超越先发出的非写入指令。

举例来说，假设当有一写入指令REQ1要经由对等运算指令路径370传送到指令下行单元322时，若有一个先发出的非写入指令REQ0的等待时间超过预设时间T，则利用对等运算指令路径370使得写入指令REQ1的优先序超前非写入指令REQ0。如此可以避免写入指令REQ1以及非写入指令REQ0互相等待而造成系统死机。

而指令下行单元322中的下行排程器3221分别接收由处理单元310以及指令上行单元321传送过来的指令，而后利用下行写入指令译码器3222以及下行非写入指令译码器3223分别译码写入指令以及非写入指令，之后再分别储存于一下行写入指令暂存器3224以及一下行非写入指令暂存器3225中。最后再传送到对应的周边元件330~360。

图5为本发明指令存取的方法流程图。首先，当收到指令REQ{REQ_P2P_WR/REQ_NONP2P_WR/REQ_P2P_RD/REQ_NONP2P_RD}时，基板地址范围译码单元331~361译码指令REQ以判断指令REQ为对等运算指令(REQ_P2P)或是非对等运算指令(REQ_NONP2P)，并判断指令REQ为写入指令

(REQ_P2P_WR/REQ_NONP2P_WR) 或是非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) (步骤S501)。接着分别将写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 储存于元件写入暂存器 332~362 中，非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) 储存于元件非写入暂存器 333~363 中 (步骤S502)。上行写入指令裁决端口 3211 以及上行非写入指令裁决端口 3212 分别决定元件写入暂存器 332~362 中的写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 以及元件非写入暂存器 333~363 中的非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) 的先后顺序 (步骤S503)。并分别将写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 储存到上行写入暂存器 3213，以及将非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) 储存于上行非写入暂存器 3214 中 (步骤S504)。接着，上行排程器 3215 依序接收储存于上行写入暂存器 3213 以及上行非写入暂存器 3214 中的写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 以及非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD)，并判断是否为对等运算指令 (REQ_P2P) (步骤S505)。其中若指令为非对等运算指令 (REQ_NONP2P)，则传送到处理单元 310 (步骤S506)。若指令为对等运算指令 (REQ_P2P_WR/REQ_P2P_RD)，则经由对等运算指令路径 370 传送到指令下行单元 322 (步骤S507)。下行排程器 3221 依序接收由处理单元 310 以及指令上行单元传送过来的指令，并将写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 传送到下行写入指令译码器 3222，以及将非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) 传送到下行非写入指令译码器 3223 (步骤S508)。下行写入指令译码器 3222 以及一下行非写入指令译码器 3223 分别译码写入指令

(REQ_P2P_WR/REQ_NONP2P_WR) 以及非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD)(步骤509)。接着, 将译码后的写入指令 (REQ_P2P_WR/REQ_NONP2P_WR) 以及非写入指令 (REQ_P2P_RD/REQ_NONP2P_RD) 依序储存于下行写入指令暂存器3224以及下行非写入指令暂存器3225中(步骤S510)。最后将处理完的指令传送到对应的周边元件330~360(步骤S511)。

图6所示为本发明中指令排程的方法流程图。首先上行排程器3215收到由上行写入暂存器3213或是上行非写入暂存器3214所传送过来的一个指令REQ_1(步骤S601)。接着判断指令REQ_1是否为对等运算指令(步骤S602)。如果指令REQ_1不是对等运算指令, 则将指令REQ_1传送到处理单元310(步骤S603)。如果指令REQ_1是对等运算指令, 判断指令REQ_1之前是否有一个非写入指令REQ_0的等待时间超过一预定时间T(步骤S604)。如果没有, 则将指令REQ_1经由对等运算指令路径传送到下行排程器3221(步骤S605)。如果有, 则判断指令REQ_1是否为一个写入指令(步骤S606)。如果不是, 则跳至步骤S605, 将指令REQ_1经由对等运算指令路径传送到下行排程器3221。如果是, 则使得指令REQ_1的优先序超前非写入指令REQ_0(步骤S607)。

举例来说, 若周边元件330发出一对等运算读取指令REQ_P2P_RE要读取周边元件340上的数据, 而周边元件340发出一对等运算写入指令REQ_P2P_WR要将数据写入周边元件350。本发明的上行排程器3215会侦测周边元件330所发出的对等运算读取指令REQ_P2P_RE的等待时间, 若等待时间超过预设时间T, 则让周边元件340所发出的对等运算写入指令REQ_P2P_WR先经由对等运算指令路径370送到指令下行单元322, 而不需要等待周边元件330发出的对等运算读取指令REQ_P2P_RE。

此外, 若欲达成更高的效率, 对等运算指令路径370更可分为

二路径,包括一对等运算存储器写入指令路径371(图中未示出),用以专门传送对等运算存储器写入指令REQ_P2P_WR;以及一对等运算存储器非写入指令路径372(图中未示出),用以专门传送对等运算存储器非写入指令REQ_P2P_RD。

本发明上述实施例所揭露的芯片320可为一北桥芯片组,亦可为一南桥芯片组。本发明指令排程的方法以及相关装置可有效地避免系统死机的问题,而使得整个系统能够顺利地运作。

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。

附图中符号的简单说明如下:

10、20: 北桥芯片组

111~114、211~214: 暂存器

120: 裁决端口

130: 译码器

140、250: 中央处理器

121~124、221~224、334~364: 周边装置

220: 上行裁决端口

230: 上行暂存器

240: 上行译码器

260、440: 对等运算指令路径

270: 下行裁决端口

280: 下行译码器

290: 下行暂存器

300: 快速周边连接装置指令存取装置

310: 处理单元

- 320: 芯片
- 321: 指令上行单元
- 3211: 上行写入指令裁决端口
- 3212: 上行非写入指令裁决端口
- 3213: 上行写入指令暂存器
- 3214: 上行非写入指令暂存器
- 3215: 上行排程器
- 322: 指令下行单元
- 3221: 下行排程器
- 3222: 下行写入指令译码器
- 3223: 下行非写入指令译码器
- 3224: 下行写入指令暂存器
- 3225: 下行非写入指令暂存器
- 151 ~ 154、251 ~ 254、330 ~ 360: 周边元件
- 331、341、351、361: 基板地址范围译码单元
- 332、342、352、362: 元件写入暂存器
- 333、343、353、363: 元件非写入暂存器
- S501~S511、S601~S607: 方法流程

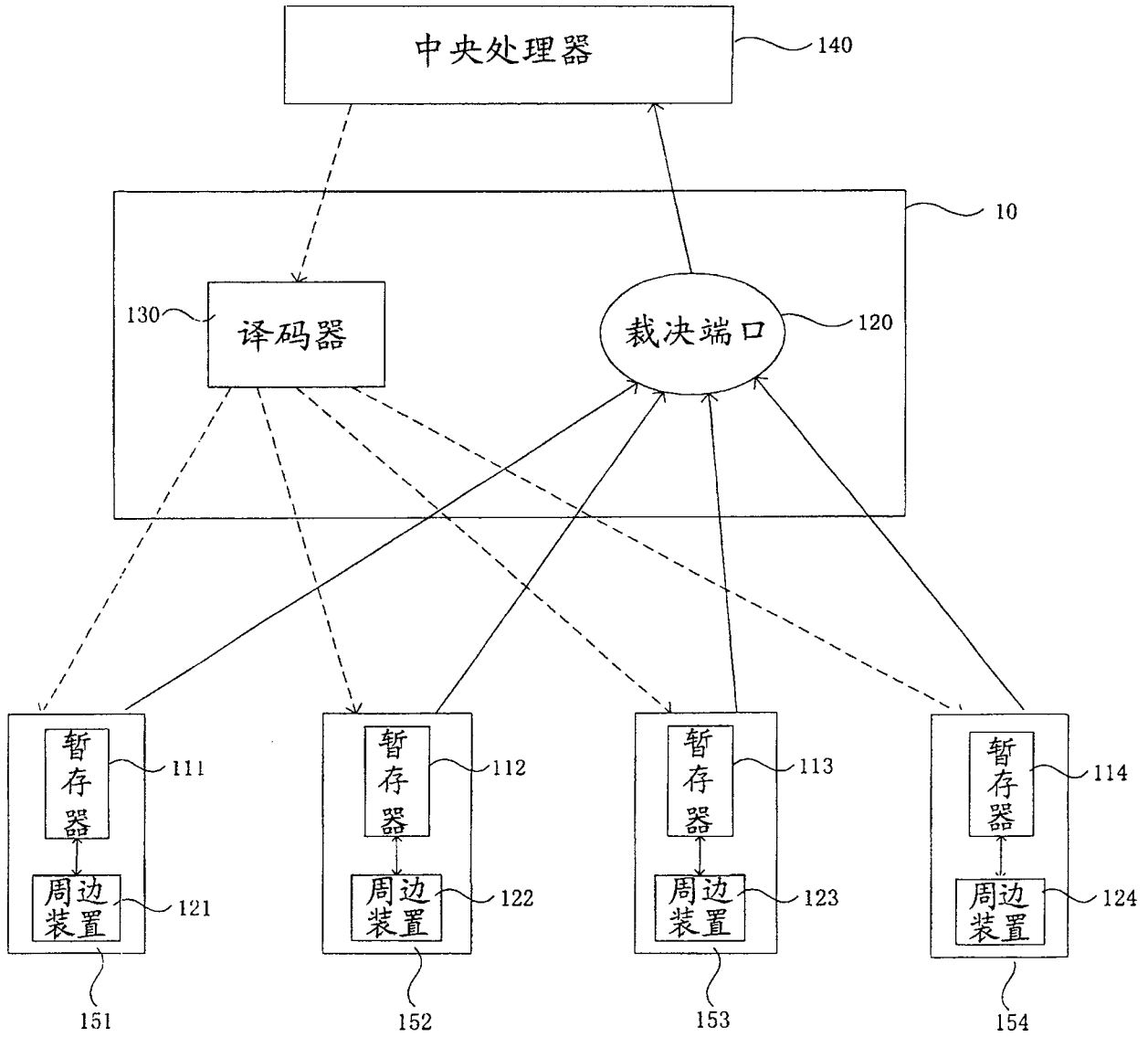


图 1

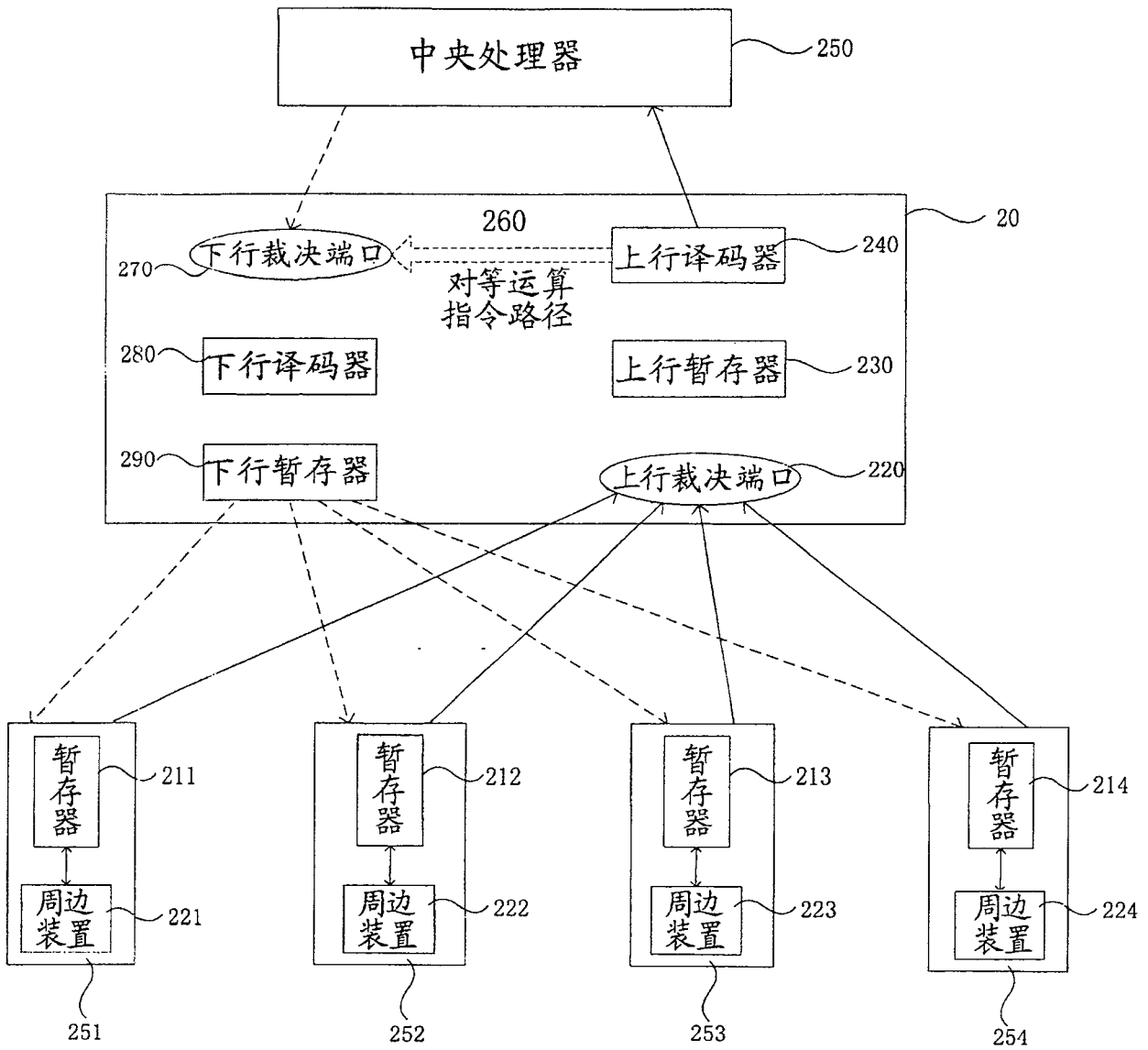


图 2

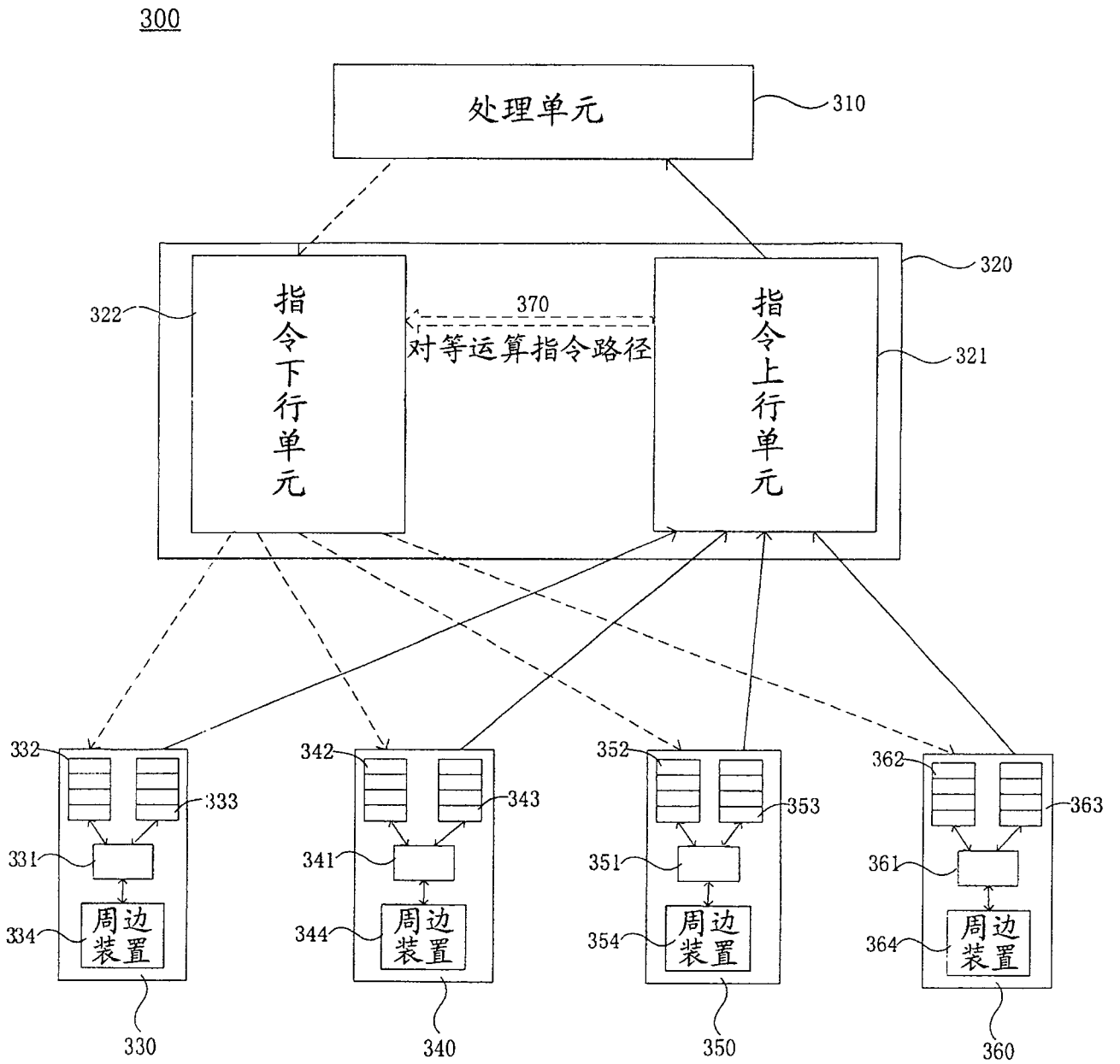


图 3

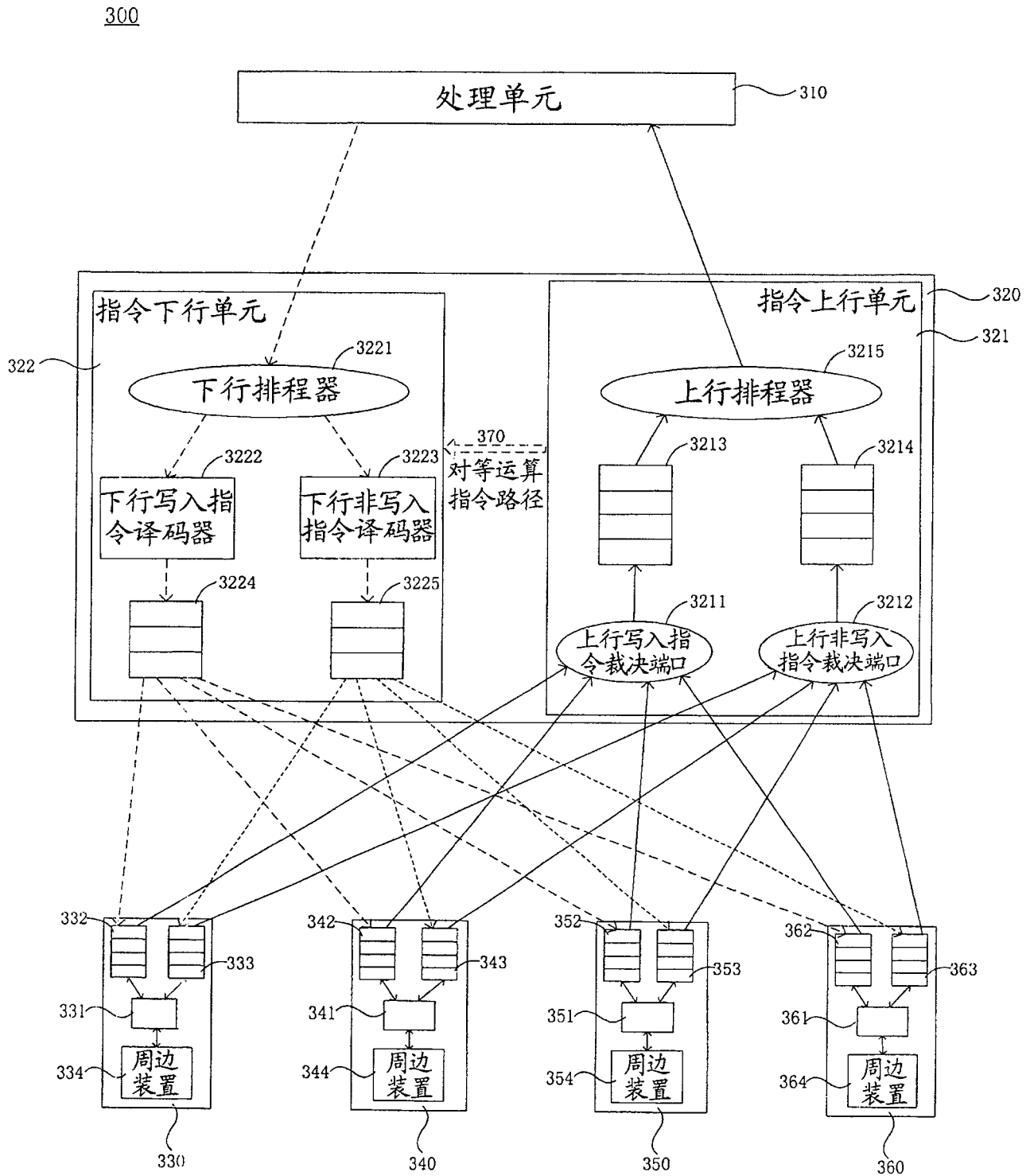


图 4

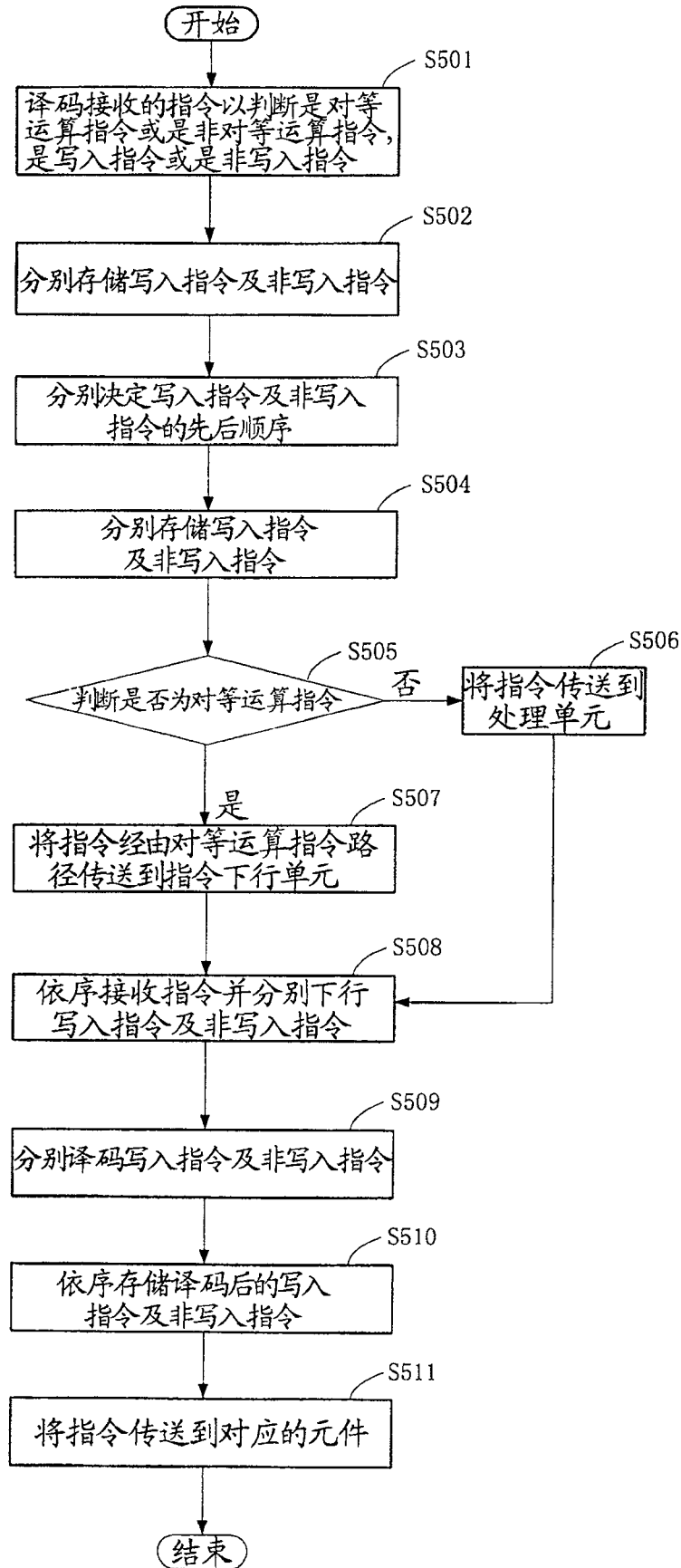


图 5

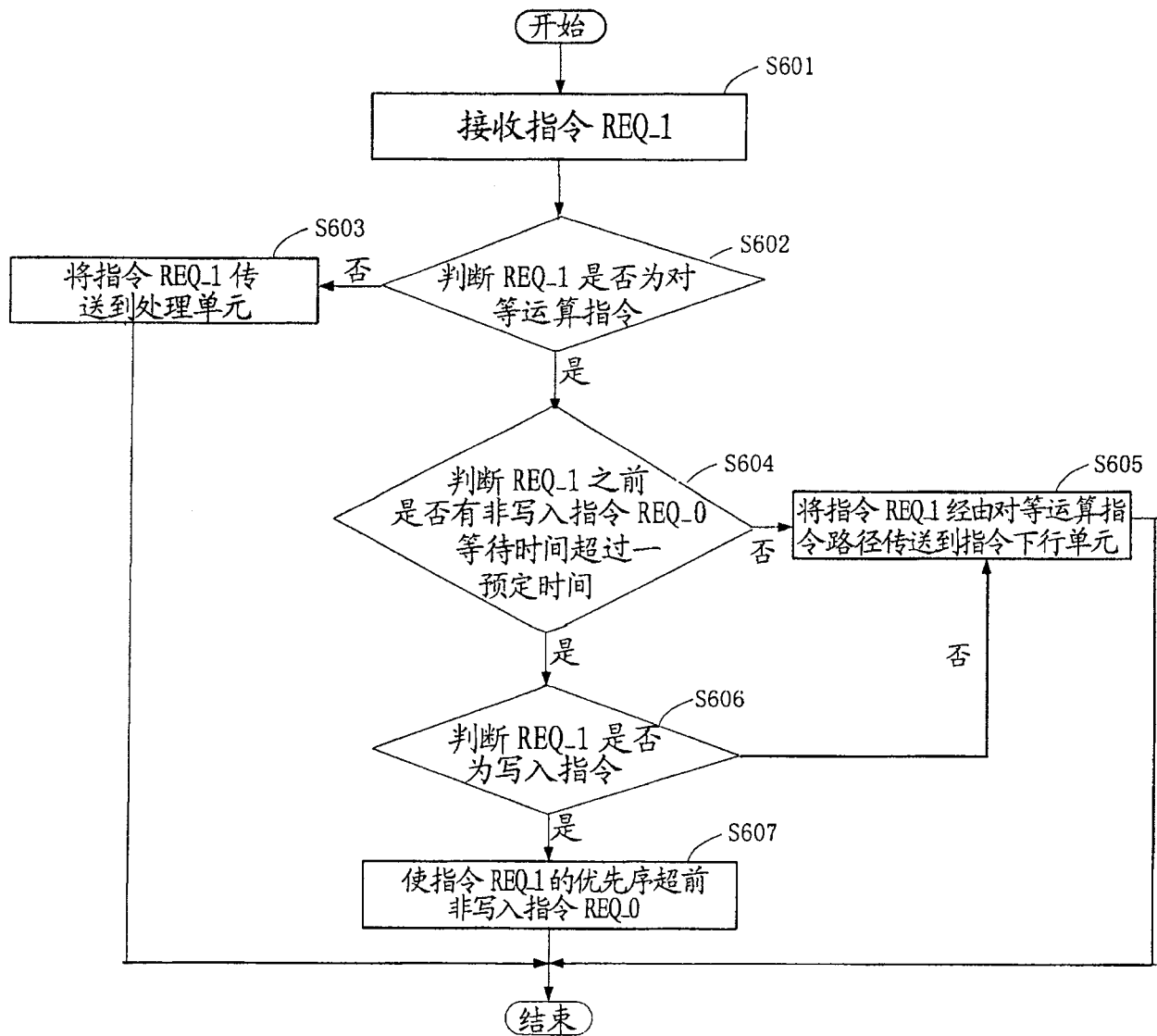


图 6