[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 13/18 (2006.01)

[12] 发明专利说明书

专利号 ZL 200610066558.5

[45] 授权公告日 2009 年 6 月 24 日

[11] 授权公告号 CN 100504825C

[22] 申请日 2006.3.30
[21] 申请号 200610066558.5
[30] 优先权
[32] 2005. 3.30 [33] JP [31] 2005 - 099420
[73] 专利权人 佳能株式会社
地址 日本东京都
[72] 发明人 石川尚
[56] 参考文献
US2003/0167294A1 2003.9.4
CN1403913A 2003.3.19
US5761445A 1998.6.2
US6804736B2 2004.10.12
US6246256B1 2001.6.12
US6839784B1 2005.1.4
审查员 何明伦

[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 权鲜枝

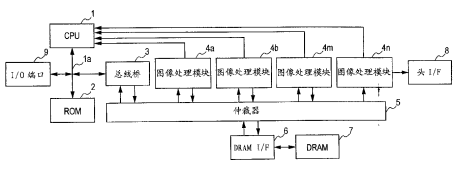
权利要求书 3 页 说明书 22 页 附图 8 页

[54] 发明名称

仲裁器和控制仲裁器的方法以及信息处理装置

[57] 摘要

一种仲裁器和控制仲裁器的方法以及信息处理装置。 在包括多个模块和仲裁该多个模块的总线访问请求的第一仲裁器的信息处理装置中，该多个模块中的至少一个包括多个子模块和第二仲裁器，该第二仲裁器仲裁该多个子模块的总线访问请求，并将该多个子模块的总线访问请求中的至少一个发送到第一仲裁器。 第一仲裁器将优先权赋予发送很多总线访问请求的模块或进行先前总线访问的模块，并限制同一模块的连续访问的数量，以控制该多个模块访问总线的优先权。 第二仲裁器根据每个子模块的缓冲器的空闲状态或访问类型来控制该多个子模块访问总线的优先权，由此可以仲裁该多个模块的总线访问请求，从而增加总线使用效率。



1. 一种仲裁器，用来仲裁多个模块对总线的访问，该仲裁器包括：

请求队列，用来堆叠从该多个模块中的每一个发送的至少一个总线访问请求；

检测单元，用来检测堆叠在该请求队列上的所述至少一个总线访问请求的数量；

识别单元，用来从该多个模块中识别进行紧前一个总线访问的模块；以及

控制单元，用来根据该多个模块中的每一个的总线访问请求的数量和识别结果，来控制访问总线的优先权。

2. 根据权利要求1所述的仲裁器，其特征在于，该控制单元将优先权赋予该进行紧前一个总线访问的模块。

3. 根据权利要求1所述的仲裁器，其特征在于，该仲裁器还包括计数单元，该计数单元用来对模块连续访问总线的数量进行计数，其中，在该连续访问的数量达到预定数量的情况下，所述控制单元禁止该连续访问的数量达到所述预定数量的模块发送总线访问请求。

4. 根据权利要求3所述的仲裁器，其特征在于，在与进行了紧前一个访问的第二总线主控器不同的第一总线主控器进行访问的情况下，或者在一个或更少的总线主控器发送总线访问请求的情况下，该计数单元将该连续访问的数量复位为0。

5. 根据权利要求1所述的仲裁器，其特征在于，所述控制单元将优先权赋予比其它模块发送更多总线访问请求的模块的总线访问。

6. 根据权利要求5所述的仲裁器，其特征在于，当从该多个模块中的一个发送的至少一个总线访问请求的数量与从该多个模块中的另一个发送的至少一个总线访问请求的数量相同时，所述

控制单元将优先权赋予以低频率产生所述至少一个总线访问请求的模块进行的总线访问。

7. 根据权利要求6所述的仲裁器，其特征在于，当该多个模块中的一个的该至少一个总线访问请求的数量和该至少一个总线访问请求的频率与该多个模块中的另一个的相同时，所述控制单元将优先权赋予以短的等待时间发送总线访问请求的模块。

8. 一种信息处理装置，包括：

多个模块；以及

第一仲裁单元，用来仲裁从该多个模块发送的总线访问请求，

其中，该第一仲裁单元包括：

检测单元，用来针对该多个模块中的每一个检测所述总线访问请求；

识别单元，用来从该多个模块中识别进行紧前一个总线访问的模块；以及

控制单元，用来根据该多个模块中的每一个的总线访问请求的数量和识别结果，来控制该多个模块访问总线的优先权，以及

其中，至少一个模块包括：

多个子模块；以及

第二仲裁单元，用来仲裁从该多个子模块发送的总线访问请求，并将从该多个子模块发送的总线访问请求中的至少一个发送到该第一仲裁单元。

9. 一种控制仲裁器的方法，该仲裁器用来仲裁多个模块对总线的访问，该方法包括以下步骤：

检测从该多个模块中的每一个发送的至少一个总线访问请求的数量，该总线访问请求被堆叠在堆叠从该多个模块的每一个发送的总线访问请求的请求队列上；

从该多个模块中识别进行紧前一个总线访问的模块；以及

根据该多个模块中的每一个的总线访问请求数量和识别结果，来控制对总线访问的优先权。

仲裁器和控制仲裁器的方法以及信息处理装置

技术领域

本发明涉及一种用来仲裁由多个模块进行的总线访问的装置和用于控制该装置的方法。

背景技术

当访问连接到动态随机存取存储器（DRAM）的存储器总线的多个总线主控器（bus master）每个都发送总线使用请求时，用来仲裁总线访问的仲裁器将该存储器总线使用权赋予该总线主控器中的一个，以控制（仲裁）总线使用权。在过去，从硬件的观点将总线使用权的优先权赋予总线主控器。因此，当同时从多个总线主控器发送总线使用请求时，总线仲裁器将总线使用允许信号发送到总线主控器中预定的一个，其通常是具有高优先权的总线主控器。随后，将总线使用权赋予该具有高优先权的总线主控器。例如，日本特开平09-062579号公报公开了上述技术。

因此，如果从具有高优先权的总线主控器频繁地发送总线使用请求，那么具有高优先权的总线主控器获得总线使用权的比率增加。在这种情况下，具有低优先权的总线主控器难以获得总线使用权。

因此，通过限制下一个总线使用请求的接收，直到将总线使用权赋予每一个接收到的总线使用请求，具有低优先权的总线主控器就可以获得总线使用权。然而，当大量总线主控器发送总线使用请求时，具有高优先权的总线主控器进行的访问数几乎与具有低优先权的总线主控器进行的访问数相同。

此外，如果当使用能成组传送（burst-transfer-capable）的总线和/或连接到DRAM等的存储器总线时，频繁地将总线使用权

从一个总线主控器转移到另一个总线主控器，则增加了地址设置的开销，并降低了总线使用效率。

此外，当通过单个仲裁器专门执行总线使用权的仲裁时，由于总线主控器数量的增加，仲裁处理变得复杂，电路尺寸增加，并且总线的高速操作性降低。

发明内容

本发明允许总线使用权的动态控制，以提高总线的使用效率。

此外，本发明允许以分布方式仲裁总线使用权，防止由于总线主控器的数量的增加而引起的仲裁器电路尺寸的增加，以及维持总线的高速操作性。

根据本发明的一个方面，提供一种仲裁器，用来仲裁多个模块对总线的访问，该仲裁器包括：检测单元，用来检测设置在该多个模块的每个中的缓冲器的空闲空间状态以存储数据；以及控制单元，用来根据所述每个缓冲器的空闲空间状态来控制该多个模块访问总线的优先权。

根据本发明的另一方面，提供一种仲裁器，用来仲裁多个模块对总线的访问，该仲裁器包括：请求队列，用来堆叠从该多个模块中的每一个发送的至少一个总线访问请求；检测单元，用来检测堆叠在该请求队列上的所述至少一个总线访问请求的数量；识别单元，用来从该多个模块中识别进行紧前一个总线访问的模块；以及控制单元，用来根据该多个模块中的每一个的总线访问请求的数量和识别结果，来控制访问总线的优先权。

根据本发明的另一方面，提供一种信息处理装置，包括：多个模块；以及第一仲裁单元，用来仲裁从该多个模块发送的总线访问请求，其中，该第一仲裁单元包括：检测单元，用来针对该多个模块中的每一个检测所述总线访问请求；识别单元，用来从

该多个模块中识别进行紧前一个总线访问的模块；以及控制单元，用来根据该多个模块中的每一个的总线访问请求的数量和识别结果，来控制该多个模块访问总线的优先权，以及其中，至少一个模块包括：多个子模块；以及第二仲裁单元，用来仲裁从该多个子模块发送的总线访问请求，并将从该多个子模块发送的总线访问请求中的至少一个发送到该第一仲裁单元。

根据本发明的另一方面，提供一种信息处理装置，包括：多个模块；以及第一仲裁单元，用来仲裁从该多个模块发送的总线访问请求，其中，至少一个模块包括：多个子模块；以及第二仲裁单元，用来仲裁从该多个子模块发送的总线访问请求，并将从该多个子模块发送的总线访问请求中的至少一个发送到该第一仲裁单元，以及其中，该第二仲裁单元包括：检测单元，用来检测设置在该多个子模块的每一个中的缓冲器的空闲空间状态以存储数据；以及控制单元，用来根据所述每个缓冲器的空闲空间状态，来控制该多个子模块访问总线的优先权。

根据本发明的另一方面，提供控制仲裁器的方法，该仲裁器用来仲裁多个模块对总线的访问，该方法包括以下步骤：检测设置在该多个模块的每一个中的缓冲器的空闲空间状态以存储数据；以及根据所述每个缓冲器的空闲空间状态来控制访问总线的优先权。

根据本发明的另一方面，提供一种控制仲裁器的方法，该仲裁器用来仲裁多个模块对总线的访问，该方法包括以下步骤：检测从该多个模块中的每一个发送的至少一个总线访问请求的数量，该总线访问请求被堆叠在堆叠从该多个模块的每一个发送的总线访问请求的请求队列上；从该多个模块中识别进行紧前一个总线访问的模块；以及根据该多个模块中的每一个的总线访问请求数量和识别结果来控制对总线访问的优先权。

根据本发明的另一方面，提供一种存储在计算机可读存储介质上的计算机可执行的处理方法，该计算机可执行的处理方法执行上述方法。

通过以下（参考附图）对典型实施例的说明，本发明的其它特征将变得很明显。

附图说明

图1是示出根据本发明第一实施例的图像处理装置的结构的例子的框图；

图2是详细示出图像处理模块的结构的例子的框图；

图3是示出在子模块之间执行数据传送的时序图；

图4是示出图像处理模块中的仲裁器的操作算法的流程图；

图5是示出图像处理装置中的仲裁器的操作算法的流程图；

图6是示出用于限制连续总线访问的算法的流程图；

图7是示出根据本发明第二实施例的图像处理模块的结构的例子的框图；

图8是示出根据本发明第二实施例的仲裁器的操作算法的流程图。

具体实施方式

以下，参考附图对本发明的典型实施例进行详细说明。

第一实施例

图像处理装置的结构

首先，对用来执行各种类型图像处理过程并向外发送图像信号的图像处理装置的例子进行说明。图1是示出根据本发明第一实施例的图像处理装置的示例结构的框图。

图1中，中央处理单元(CPU)1根据存储在只读存储器(ROM)

2中的程序，通过使用动态随机存取存储器（DRAM）7作为工作存储器，控制整个图像处理装置。此外，CPU 1通过CPU总线1a与ROM 2、总线桥3、以及输入/输出（I/O）端口9连接。

仲裁器5仲裁CPU 1通过总线桥3对DRAM 7进行的访问以及n个图像处理模块4（其中，保持表达式n≥1且n为整数）对DRAM 7进行的访问。此外，DRAM 7具有DRAM接口（I/F）6。

此外，图像处理模块4中的一个，例如，图1中所示的图像处理模块4n，通过头接口（I/F）8连接到喷墨打印机的打印头。

图1中，CPU 1和图像处理模块4共享DRAM 7。然而，可以将CPU 1专用的随机存取存储器（RAM）（未示出）连接到CPU总线1a，以维持并增加图像处理装置的性能。

处理操作

CPU 1接收从I/O端口9发送来的用于根据存储在ROM 2中的程序进行处理的图像数据，并通过总线桥3、仲裁器5、和DRAM I/F 6将该图像数据存储在DRAM 7中。接着，CPU 1设置图像处理模

块4a的配置寄存器，使得图像处理模块4a工作。

图像处理模块4a执行预定处理。在读取或写入设置在配置寄存器中的用于处理的数据后，图像处理模块4a产生中断并将其发送到CPU 1，以通知CPU 1该处理被完成。

一旦接收到该中断，CPU 1就分析该中断产生的原因。当完成了由图像处理模块4a执行的读取处理时，CPU 1设置下一个用于处理的数据，并使图像处理模块4a继续执行处理。此外，当完成了由图像处理模块4a执行的写入处理时，CPU 1设置存储下一个用于处理的数据的位置，使图像处理模块4a继续执行处理，为下一个图像处理模块4b设置配置寄存器，并使图像处理模块4b工作。

图像处理模块4b执行预定处理。在读取或写入设置在配置寄存器中的用于处理的数据后，图像处理模块4b产生中断并将其发送到CPU 1，以通知CPU 1该处理结束。

一旦接收到该中断，CPU 1就分析该中断产生的原因。当完成了由图像处理模块4b执行的读取处理时，CPU 1设置下一个用于处理的数据，并使图像处理模块4b继续执行处理。此外，当完成了由图像处理模块4b执行的写入处理时，CPU 1设置存储下一个用于处理的数据的位置，使图像处理模块4b继续执行处理，为下一个图像处理模块4c设置配置寄存器，并使图像处理模块4c工作。

这样，在完成了由预定图像处理模块执行的前面的处理后，立即起动下一个图像处理模块，并将用于处理的数据发送到该下一个图像处理模块。通过重复执行上述操作，可以形成以图像处理模块为单位的流水线。

当图像处理模块4m完成了执行上述处理时，产生预定量或以上的位图数据。随后，CPU 1启动打印机引擎（未示出），使图像

处理模块4n与从打印机引擎发送的同步信号同步开始执行处理，并将该位图数据通过头I/F 8发送到打印机引擎，使得打印机引擎打印该位图数据的图像。

图像处理模块的结构

图2是详细示出图像处理模块4的示例结构的框图。图像处理模块4包括读缓冲器10、m个子模块11（其中，保持表达式m≥1且m为整数）、写缓冲器12、仲裁器13、读地址发生器14、中断控制器15以及写地址发生器16。

根据图像处理模块4的配置寄存器上的设置，CPU 1对读地址发生器14设置关于读起始地址和/或读结束地址的信息、以及读使能（enable）信号Ren。此外，CPU 1对写地址发生器16设置关于写起始地址和/或写结束地址的信息、以及写使能信号Wen。

仲裁器13检测读缓冲器10的空闲空间Rp和读地址发生器14的使能信号Ren。如果读地址有效（Ren='1'），并且可将数据存储在读缓冲器10中（Rp≥Rn），则发出读请求（PREQ='1'、PNRW='0'、PNUM=Rn、和PADD=Rad），并将其发送到仲裁器5。

如果写缓冲器12的数据累积数Wp的值大于等于预定字数（Wp≥Wn），则仲裁器13检测写地址发生器16的使能信号Wen。如果写地址有效（Wen='1'），则仲裁器13发出写请求（PREQ='1'、PNRW='1'、PNUM=Wn、和PADD=Wad），并将其发送到仲裁器5。

一旦接收到从图像处理模块4发送的请求信号PREQ，仲裁器5就根据由PNRW代表的信息判断请求信号PREQ是表示读请求还是写请求，并根据由PNUM代表的信息检测字数，以及根据由PADD代表的信息检测读和/或写地址。如果此时从CPU 1和任一其它图像处理模块4都没有发送请求，则仲裁器5通过DRAM I/F 6开始访问DRAM 7的地址。如果通过DRAM I/F 6接收到请求，则

仲裁器5将接收信号PACK返回到作为请求源的图像处理模块4。另一方面，当从CPU 1和任一其它图像处理模块4发送请求时，仲裁器5按照优先权的降序接收该请求。

在接收信号PACK被发送且该请求为读请求的情况下，仲裁器13将接收信号Rack发送到作为请求源的读地址发生器14。当接收信号PACK被发送且该请求为写请求时，仲裁器13将接收信号Wack发送到作为请求源的写地址发生器16。

一旦接收到接收信号Rack，读地址发生器14就产生下一个地址。如果所发出的请求所针对的地址是读结束地址，则读地址发生器14复位读使能信号Ren，并将读结束信号Rend发送到中断控制器15。一旦接收到接收信号Wack，写地址发生器16就产生下一个地址。如果所发出的请求所针对的地址是写结束地址，则写地址发生器16复位写使能信号Wen，并将写结束信号Wend发送到中断控制器15。

中断控制器15可以通过使用配置寄存器设置读结束中断屏蔽和写结束中断屏蔽。如果每个中断屏蔽上的设置代表允许中断，则中断控制器15根据读结束信号Rend和/或写结束信号Wend产生中断信号INT，并将该中断信号INT发送到CPU 1。

一旦接收到中断信号INT，CPU 1就读取关于中断控制器15的状态信息。如果因为完成了读处理而产生该中断信号，则CPU 1复位读结束中断屏蔽，并取消该中断信号INT。如果需要继续该处理，则CPU 1再次设置读起始地址和读结束地址，设置读使能信号Ren，并设置读结束中断屏蔽。此外，如果因为完成了写处理而产生该中断信号，则CPU 1复位写结束中断屏蔽，并取消该中断信号INT。如果需要继续该处理，则CPU 1再次设置写起始地址和写结束地址，设置写使能信号Wen，并设置写结束中断屏蔽。

接着，当从DRAM 7读取数据时，仲裁器5将DRAM数据有效

信号PVALID发送到作为请求源的图像处理模块4。图像处理模块4的仲裁器13将数据有效信号Rvalid发送到读缓冲器10。读缓冲器10在数据有效信号Rvalid被设置期间的时间段存储DRAM数据输出信号PDIN上的数据。从而，将DRAM 7上的数据存储在读缓冲器10中。

另一方面，当将数据写入DRAM 7时，仲裁器5将DRAM数据有效信号PVALID发送到图像处理模块4，该图像处理模块4是在将该数据写入DRAM 7时的请求源。作为请求源的图像处理模块4的仲裁器13将数据有效信号Wvalid发送到写缓冲器12。写缓冲器12发送待写入DRAM 7的数据，作为在数据有效信号Wvalid被设置期间的时间段的DRAM数据输入信号PDOUT。从而，将写缓冲器12上的数据存储在DRAM 7中。

当准备由子模块11a执行的处理所需的每一数据项时，读缓冲器10设置有效信号valid_0。否则，读缓冲器10使有效信号valid_0复位。

当没有设置从子模块11a发送的存储请求信号stall_0时，读缓冲器10与时钟信号同步地向外发送存储在其中的数据。然而，当设置了存储请求信号stall_0时，读缓冲器10不更新存储在其中的数据。

子模块11a仅接收设置有有效信号valid_0的数据。如果子模块11a难以接收此数据，则子模块11a设置存储请求信号stall_0，并保持从读缓冲器10发送的信号。如果不需要重新排列输入数据项，则可以将读缓冲器10形成为先进先出（FIFO）存储器。同样地，如果不需要重新排列输出数据项，则可以将写缓冲器12形成为FIFO存储器。

图像处理模块4包括至少一个用来执行图像处理的子模块11。如果具有两个子模块11，则在子模块11之间执行与上述相同的操

作（即，通过使用有效信号valid_x和存储请求信号stall_x的信号交换（hand shaking）），使得在子模块11之间发送和/或接收数据data_x。

在子模块之间执行的数据传送

图3是示出在子模块11之间执行的数据传送的时序图。当可以发送数据时，数据发送侧的子模块11与时钟信号clk的上升沿同步地设置数据信号d1和有效信号valid（T1）。如果接收侧的子模块11没有与下一个信号的上升沿同步地设置存储请求信号stall，则判断为接收到数据信号d1。然后，如果可以发送下一数据，则数据发送侧的子模块11设置数据信号d2和有效信号valid（T2）。如果难以发送下一数据，则数据发送侧的子模块11复位有效信号valid（T3）。

当接收侧的子模块11在下一个时钟信号的上升沿设置存储请求信号stall时，则判断为该数据信号未被接收并保持数据信号d5和有效信号valid（T7）。此外，如果没有设置有效信号valid，即使通过接收侧的子模块11设置了存储请求信号stall（T8），数据信号d5也是无效的。在这种情况下，接收侧的子模块11发送作为下一个有效数据的数据信号d6，并设置有效信号valid，而无需保持数据信号d5和有效信号valid（T9）。也就是说，当没有设置有效信号valid时，忽略存储请求信号stall。

当数据接收侧的子模块11可以接收数据时，子模块11与时钟信号clk的上升沿同步地接收对其设置了有效信号valid的数据信号（T1、T2、T4和T5）。如果数据接收侧的子模块11难以接收数据，则上述子模块11设置存储请求信号stall，并使发送侧的子模块11保持数据信号d5和有效信号valid（T6）。然后，当数据接收侧的子模块11能够接收数据时，其复位存储请求信号stall并接收发送到其的数据信号d5（T7）。

如果写缓冲器12具有空闲空间，则其存储当子模块11设置有效信号valid_n时所获得的数据信号data_n。如果写缓冲器12没有空闲空间，则其设置存储请求信号stall_n，并使子模块11保持输出信号。

图像处理模块中的仲裁器13的操作算法

图4是示出仲裁器13的操作算法的流程图。以下，将请求队列上所累积的请求的数量确定为Pp，将执行请求队列上所累积的请求时写缓冲器12上所累积的数据项的数量（评估值）确定为Pw，而将执行请求队列上所累积的请求时读缓冲器10的空闲空间（评估值）确定为Pr。当仲裁器5接收到请求（PACK='1'）时，Pp的值递减1。以下，假定读请求产生的频率大于写请求产生的频率。此外，按照对总线访问的请求的产生频率的升序，对每个模块检测后面将进行说明的关于缓冲器的空闲空间的状态的信息。

当写缓冲器12上所累积的数据项的数量的评估值Pw大于等于预定的字数Wn（Pw≥Wn），且写地址为有效（Wen='1'）时，可以由表达式Wreq='1'代表写请求Wreq。此外，当读缓冲器10的空闲空间的评估值Pr大于等于预定的字数Rn（Pr≥Rn），且读地址Ren为有效（Ren='1'）时，可以由表达式Rreq='1'代表读请求Rreq。

首先，在步骤S201，判断是否由表达式Wreq='1'代表写请求Wreq，且由表达式Pw≥Wth代表评估值Pw和预定值Wth间的关系或者请求队列上所累积的上一请求是由表达式ID1=IDw代表的写请求。如果是，则流程进入步骤S205，在该步骤，在请求队列上累积该写请求。

如果未满足上述条件，则流程进入步骤S202，在该步骤，判断是否由表达式Rreq='1'代表读请求，且由表达式Pr≥Rth代表评估值Pr和预定值Rth之间的关系或者请求队列上所累积的上一请求是由表达式ID1=IDr代表的读请求。如果是，则流程进入步骤

S206，在该步骤，在请求队列上累积该读请求。

在步骤S203，如果未满足上述两个条件，且由表达式Rreq='1'代表读请求Rreq，则流程进入步骤S206，在该步骤，在请求队列上累积该读请求Rreq。如果不是由表达式Rreq='1'代表读请求Rreq，则流程进入步骤S204。在步骤S204，如果由表达式Wreq='1'代表写请求Wreq，则在步骤S205，在请求队列上累积该写请求Wreq。

在步骤S205，当在请求队列上累积写请求Wreq时，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为写请求识别码IDw。同时，从评估值Pw减去写数据数Wn，从而更新评估值。此外，表示在请求队列上累积的请求的数量的请求数Pp的值被递增1。

此外，在步骤S206，当在请求队列上累积读请求Rreq时，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为读请求识别码IDr。同时，从评估值Pr中减去读数据数Rn，从而更新评估值Pr。此外，表示在请求队列上累积的请求的数量的请求数Pp的值被递增1。

在完成上述处理过程后，处理返回到步骤S201，以便再次执行上述处理过程。不需要设置表示请求队列上所累积的请求的数量的请求数Pp的上限，这是因为每一缓冲器的容量和仲裁器5的序列等对请求数Pp施加有限制。然而，如果根据系统配置需要设置请求数Pp的上限，则可以对请求数Pp设置最大值，使得当请求数Pp的值达到最大值时，表达式Rreq=Wreq=0成立。

图像处理装置的仲裁器5的操作算法

图5是示出仲裁器5的操作算法的流程图。在下面的说明中，三个图像处理模块M1、M2和M3、引擎处理模块M4以及总线桥B0与仲裁器5相连接。将最高优先权赋予引擎处理模块M4，以执

行实时控制。将第二高优先权赋予总线桥B0。三个图像处理模块M1、M2和M3的优先权彼此相同。因此，可以用下面的表达式表示上述模块的优先权：

M4>B0>M1、M2和M3。

首先，在步骤S211，仲裁器5判断具有最高优先权的引擎处理模块M4是否发送对总线使用权的请求req4，其由表达式req4='1'示出。如果表达式req4='1'成立，则在步骤S216，仲裁器5接收请求req4，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为引擎处理模块M4的请求识别码ID_4。然后，仲裁器5将接收信号PACK发送到引擎处理模块M4，处理返回到步骤S211。

在步骤S211，如果引擎处理模块M4没有发送请求req4，那么在步骤S212，仲裁器5判断总线桥B0是否发送请求req0，其由表达式req0='1'示出。如果表达式req0='1'成立，则流程进入步骤S217，在该步骤，仲裁器5接收请求req0，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为总线桥B0的请求识别码ID_0。然后，仲裁器5将接收信号PACK发送到总线桥B0，处理返回到步骤S211。

在步骤S212，如果总线桥B0没有发送请求req0，那么在步骤S213，仲裁器5判断图像处理模块M1是否发送请求req1，其由表达式req1='1'示出。如果表达式req1='1'成立，则仲裁器5判断图像处理模块M1的请求数P1的值是否是三个图像处理模块M1、M2和M3的请求数的值中最大的，其中，该请求数P1表示在请求队列上累积的请求的数量。也就是说，仲裁器5判断表达式P1=Pmax是否成立。此外，仲裁器5判断发送到其的紧前一个请求是否是从图像处理模块M1发送的请求，其由表达式ID1=ID_1示出。如果由表达式req1='1'示出判断结果，并且表达式P1=Pmax或表达式

ID1=ID_1成立，则仲裁器5接收请求req1。然后，在步骤S218，仲裁器将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为图像处理模块M1的请求识别码ID_1。然后，仲裁器5将接收信号PACK发送到图像处理模块M1，处理返回到步骤S211。

此外，如果表达式req1='0'和/或表达式P1≠Pmax成立且表达式ID1≠ID_1成立，则流程进入步骤S214，在该步骤，仲裁器5对图像处理模块M2执行如对图像处理模块M1所执行的相同的处理。也就是说，仲裁器5判断请求req2是否被发送，其由表达式req2='1'示出，和请求数P2是否是图像处理模块M1、M2和M3的请求数中最大的，其由表达式P2=Pmax示出，或者紧前一个请求是否是从图像处理模块M2发送的，其由表达式ID1=ID_2示出。如果判断结果示出为表达式req2='1'，且表达式P2=Pmax或者表达式ID1=ID_2成立，则仲裁器5接收请求req2。然后，在步骤S219，仲裁器5将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并将当前的请求识别码ID更新为图像处理模块M2的请求识别码ID_2。然后，仲裁器5将接收信号PACK发送到图像处理模块M2，处理返回到步骤S211。

如果表达式req2='0'和/或表达式P2≠Pmax成立且表达式ID1≠ID_2成立，则在步骤S215和S220，仲裁器5对图像处理模块M3执行如对图像处理模块M1和M2所执行的相同的处理，然后处理返回到步骤S211。

对于三个图像处理模块M1、M2和M3的优先权，可以将最高优先权赋予其请求数P1的值为最高的图像处理模块，或者赋予与发送紧前一个请求的图像处理模块相同的图像处理模块。

根据上述算法，部分地固定优先权顺序。因此，如果具有高优先权的总线主控器例如引擎处理模块M4和总线桥B0频繁地发

送总线使用权的请求，则该总线主控器可以独占使用该总线。特别地，根据上述配置，引擎处理模块M4的优先权高于总线桥B0的优先权。随后，在预定定时对CPU 1的请求的应答性恶化。此外，在图像处理模块M1、M2和M3的情况下，将最高优先权赋予发送紧前一个请求的图像处理模块。因此，图像处理模块中的预定的一个可以独占使用该总线。在这种情况下，设置连续访问数的上限，从而不让预定的总线主控器独占使用总线。

图6是示出用于限制对总线的连续访问数的算法的流程图。

首先，在步骤S221，检测表示当前发送请求的模块的数量的模块数Nr。接着，在步骤S222，对当前请求识别码ID的值与紧前一个请求识别码寄存器ID1的值进行比较。如果表达式Nr≤1或表达式ID1≠ID成立，那么在步骤S223，将计数器C的值复位为0。

如果表达式Nr>1和表达式ID1=ID成立，那么在步骤S224，进行是否接收信号PACK=1（即，将接收信号PACK发送到作为目标的模块）的判断。如果PACK=1，则流程进入步骤S225，在该步骤，每次将接收信号PACK发送到作为目标的模块，递增计数器C。

通过重复上述图6中所示的处理过程，通过计数器C计数由同一模块进行的连续总线访问的数量（以下称之为连续总线访问数）。在步骤S226，进行计数器C是否已经达到预定的值Cth的判断。当计数器C的值达到该预定值时，流程进入步骤S227，在该步骤，仲裁器5屏蔽从由紧前一个请求识别码寄存器ID1表示的模块发送的请求。在计数器C的值小于该预定的值时，在步骤S228，取消该请求的屏蔽。

根据上述配置，可以接收从不同于上述模块的模块发送的请求，从而限制连续总线访问数。此外，一旦将总线使用权转移到不同的模块，通过在步骤S221到S223中执行的处理过程复位计数

器C的值。

在图6中，全部上述模块共享对连续总线访问数进行计数的计数器C。然而，可以在每一个模块中都提供计数器C，使得可以由每个模块限制连续总线访问数。例如，如果将总线桥B0的限制值调整到CPU 1的高速缓存的行大小，则可以有效地更新高速缓存的值。

此外，当上述模块访问DRAM 7不同的存储体（bank）时，可以不考虑总线访问的连续性。从而，可以不进行当前使用的请求识别码ID和紧前一个请求识别码寄存器ID1之间的比较。

这样，通过根据缓冲器的空闲空间的状态、紧前一个访问、和连续访问的数量来动态地改变（控制）总线访问优先权，可以在增加总线访问连续性的同时，根据每个总线主控器的总线使用频率来仲裁总线访问。

此外，由于不仅通过整个图像处理装置的仲裁器5，而且还通过每个模块所具有的仲裁器13执行仲裁（分布的仲裁），因而可以执行适合每个模块的仲裁。例如，如果总线主控器的请求产生频率彼此不同，则按照请求产生频率的升序执行指针（pointer）评价和连续访问数评价。通过将较高的优先权赋予具有低请求产生频率的总线主控器，增加具有低请求产生频率的总线主控器获得总线使用权的比率，并实现处理均衡化。如果总线主控器的请求产生频率几乎彼此相同，则根据由此进行的写入访问，将较高的优先权赋予该总线主控器。这是因为执行写入处理所需的等待时间（latency）短于执行读取处理所需的等待时间，并且直到打开存储器总线为止的时间段较短。

第二实施例

以下，将对根据本发明的第二实施例执行的图像处理进行说明。在第二实施例中，用相同的附图标记表示与第一实施例相同

的要素，并省略对其说明。

图7是示出根据第二实施例的图像处理模块的示例结构的框图。图7中所示的图像处理模块与图2中所示的图像处理模块的不同之处在于：在图7中设置有两个读缓冲器10a和10b、两个写缓冲器12a和12b、两个读地址发生器14a和14b、以及两个写地址发生器16a和16b。

仲裁器13检测关于表示读缓冲器10a的空闲空间的缓冲器空间R0p的信息和读地址发生器14a的使能信号R0en。如果读地址为有效（R0en='1'），且读缓冲器10a可以存储数据（R0p≥R0n），则仲裁器13将读请求（PREQ='1'、PNRW='0'、PNUM=R0n和PADD=R0ad）发送到仲裁器5。同样地，仲裁器13检测关于表示读缓冲器10b的空闲空间的缓冲器空间R1p的信息和读地址发生器14b的使能信号R1en。如果读地址为有效（R1en='1'），且读缓冲器10b可以存储数据（R1p≥R1n），则仲裁器13将读请求（PREQ='1'、PNRW='0'、PNUM=R1n和PADD=R1ad）发送到仲裁器5。

如果表示在写缓冲器12a上累积的数据项的数量的累积数据数W0p大于等于预定的字数（W0p≥W0n），则仲裁器13检测从写地址发生器16b发送的使能信号W0en。当写地址为有效（W0en='1'）时，仲裁器13将写请求（PREQ='1'、PNRW='1'、PNUM=W0n和PADD=W0ad）发送到仲裁器5。同样地，如果表示在写缓冲器12b上累积的数据项的数量的累积数据数W1p大于等于预定的字数（W1p≥W1n），则仲裁器13检测从写地址发生器16b发送的使能信号W1en。当写地址为有效（W1en='1'）时，仲裁器13将所示的写请求（PREQ='1'、PNRW='1'、PNUM=W1n和PADD=W1ad）发送到仲裁器5。

图8是示出第二实施例的仲裁器13的操作算法的流程图。与第

一实施例的情况一样，当执行请求队列上累积的请求时，将写缓冲器12a上累积的数据项的数量（评估值）确定为Pw0，将写缓冲器12b上累积的数据项的数量（评估值）确定为Pw1，将当执行请求队列上累积的请求时的读缓冲器10a的空闲空间（评估值）确定为Pr0，以及将读缓冲器10b的空闲空间（评估值）确定为Pr1。当仲裁器5接收请求（PACK='1'）时，Pp的值递减1。以下，假定表达式（读缓冲器10a的请求产生频率）>（读缓冲器10b的请求产生频率）=（写缓冲器12b的请求产生频率）>（写缓冲器12a的请求产生频率）成立。此外，按照总线访问请求产生频率的升序，检测每个模块的缓冲器空闲空间状态的信息。

当写缓冲器12a上累积的数据项数量的评估值Pw0大于等于预定的字数W0n（Pw0≥W0n），且写地址为有效（W0en='1'）时，可以由表达式W0req='1'代表写缓冲器12a的写请求W0req。此外，当表示写缓冲器12b上累积的数据项数量的评估值Pw1大于等于预定的字数W1n（Pw1≥W1n），且写地址Wlen为有效（W1en='1'）时，可以由表达式W1req='1'代表写缓冲器12b的写请求W1req。

当表示读缓冲器10a的空闲空间的评估值Pr0大于等于预定的字数R0n（Pr0≥R0n），且读地址为有效（R0en='1'）时，可以由表达式R0req='1'代表读缓冲器10a的读请求R0req。此外，当表示读缓冲器10b的空闲空间的评估值Pr1大于等于预定的字数R1n（Pr1≥R1n），且读地址为有效（R1en='1'）时，可以由表达式R1req='1'代表读缓冲器10b的读请求R1req。

首先，在步骤S231，判断是由表达式W0req='1'代表写请求W0req，并且评估值Pw0是否大于等于预定值W0th或请求队列上累积的紧前一个请求是否是写缓冲器12a的写请求（ID1=IDw0）。如果是，则在步骤S239，在请求队列上累积写缓冲器12a的写请求。

如果未满足上述条件，则流程进入步骤S232，在该步骤，判断是否由表达式W1req=1代表写请求W1req，并且表示写缓冲器12b上累积的数据项数量的评估值P1w是否大于等于预定值W1th或者请求队列上累积的紧前一个请求是否是写缓冲器12b的写请求IDw1（ID1=IDw1）。如果是，则在步骤S240，在请求队列上累积写缓冲器12b的写请求。

如果未满足上述两个条件，则在步骤S233，判断是否由表达式Rreq='1'代表读请求Rreq，并且表示读缓冲器10b的空闲空间的评估值Pr1是否等于或高于预定值R1th或者请求队列上累积的紧前一个请求是否是读缓冲器10b的读请求（ID1=IDr1）。如果是，则流程进入步骤S242，在该步骤，在请求队列上累积读缓冲器10b的读请求。

如果未满足上述三个条件，则在步骤S234判断是否由表达式R0req='1'代表读请求R0req，并且表示读缓冲器10a的空闲空间的评估值Pr0是否等于或高于预定值R0th或者请求队列上累积的紧前一个请求是否是读缓冲器10a的读请求（ID1=IDr0）。如果是，则在步骤S241，在请求队列上累积读缓冲器10a的读请求。

如果未满足上述四个条件，则根据每个请求执行以下处理。在步骤S235，如果读请求R0req='1'，则在步骤S241，在请求队列上累积读缓冲器10a的读请求。在步骤S236，如果读请求R1req='1'，则在步骤S242，在请求队列上累积读缓冲器10b的读请求。在步骤S237，如果写请求W1req='1'，则在步骤S240，在请求队列上累积写缓冲器12b的写请求。在步骤S238，如果写请求W0req='1'，则在步骤S239，在请求队列上累积写缓冲器12a的写请求。

在步骤S239，当在请求队列上累积写缓冲器12a的写请求时，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1

中，并将当前的请求识别码ID更新为写请求识别码IDw0。同时，从表示写缓冲器12a上累积的数据项数量的评估值Pw0中减去写数据数W0n，从而更新评估值Pw0，并且表示请求队列上累积的数据项数量的请求数Pp递增。

当在步骤S240在请求队列上累积写缓冲器12b的写请求，以及在步骤S241和/或步骤S242在请求队列上累积读缓冲器10a和/或读缓冲器10b的读请求时，执行与上述处理相同的处理。也就是说，将当前的请求识别码ID存储在紧前一个请求识别码寄存器ID1中，并更新当前的请求识别码ID。同时，从缓冲器的评估值中减去数据数，从而更新评估值，并且表示请求队列上累积的数据项的数量的请求数Pp递增。

在完成上述处理后，处理返回到步骤S231，以便再次执行上述处理。

每个上述模块都包括两个数据路径。例如，在DRAM 7上设置有模块的误差扩散电路的误差缓冲器的情况下，设置用于图像数据的数据路径和用于误差缓冲器的数据路径。从而，总线主控器的数量从两个增加到4个，这使得难以将总线主控器连接到存储器总线。在这种情况下，在每个模块中执行上述仲裁，并将模块连接到存储器总线。因而，可以在不改变上层电路结构的情况下将所述模块连接到存储器总线。

此外，如上所述，不仅通过仲裁器5，而且还通过设置在每个模块中的仲裁器13执行仲裁（仲裁分布），这允许针对每个模块执行适合的仲裁。例如，如果总线主控器的请求产生频率彼此不同，则按照请求产生频率的升序，执行指针评价和连续访问数评价。通过将较高的优先权赋予具有低请求产生频率的总线主控器，增加了具有低请求产生频率的总线主控器获得总线使用权的比率，并实现了处理均衡化。如果总线主控器的请求产生频率几乎相同，

则将较高的优先权赋予进行写入访问的总线主控器。这是因为执行写入处理所需的等待时间和直到打开存储器总线的时间段，短于执行读取处理所需要的。

此外，如图6中所示，通过使用对连续访问数进行计数的计数器C可以限制连续访问数。计数器C的使用允许在不考虑上述产生频率的情况下，实现处理均衡化。

此外，可以设置与上述评估值Pw0、Pw1、Pr0和Pr1相比较的阈值W0th、W1th、R0th和R1th，以便执行更高级的仲裁。在这种情况下，可以按照阈值的降序设置优先权，并且当将评估值与最大阈值比较时，可以判断连续访问数。因而，可以对每个模块执行优化，使得容易地增加存储器总线的使用效率。

在上述实施例中，执行对存储器总线的访问的仲裁。然而，本发明可用于对各种类型的总线的使用权进行仲裁，而不限于存储器总线。

根据上述实施例，通过动态控制总线使用权，可以增加总线使用效率。此外，通过以分布方式执行总线使用权仲裁，可以防止由于总线主控器数量的增加而引起的电路尺寸的增加，并维持总线的高速操作性。

其它实施例

本发明可以用于包括多个装置的系统，该装置包括主计算机、接口、读取器和打印机等；或可以用于作为一个单元而形成的装置，该装置包括复印机、传真机等。

应该理解，本发明的目的还可以通过装置或系统的计算机（CPU、MPU等）来实现：提供该计算机，使得从存储用于实现上述实施例的功能的软件的程序代码的存储介质中，读取该程序代码，并执行该程序代码。在这种情况下，从存储介质读取的程序代码自身实现了上述实施例的功能，因而存储该程序代码的存

储介质构成了本发明。此外，不仅可以通过计算机读取并执行该程序代码，还可以通过计算机基于该程序代码的指令利用运行在计算机上的操作系统（OS）等执行部分或全部处理来实现上述实施例的功能。后者也是本发明的一个实施例。

在本发明的另一实施例中，可以将从存储介质读取的程序代码写入到插入计算机中的功能扩展卡或与计算机连接的功能扩展单元的存储器中。可以基于该程序代码的指令，通过该功能扩展卡或功能扩展单元的CPU等执行全部或部分实际处理，以实现上述实施例的功能。

当本发明用于上述存储介质时，该存储介质存储与上述流程图相对应的程序代码。

尽管已经参考典型实施例说明了本发明，但是应该理解，本发明不局限于所公开的典型实施例。所附权利要求的范围符合最宽的解释，以包含全部修改、等同结构和功能。

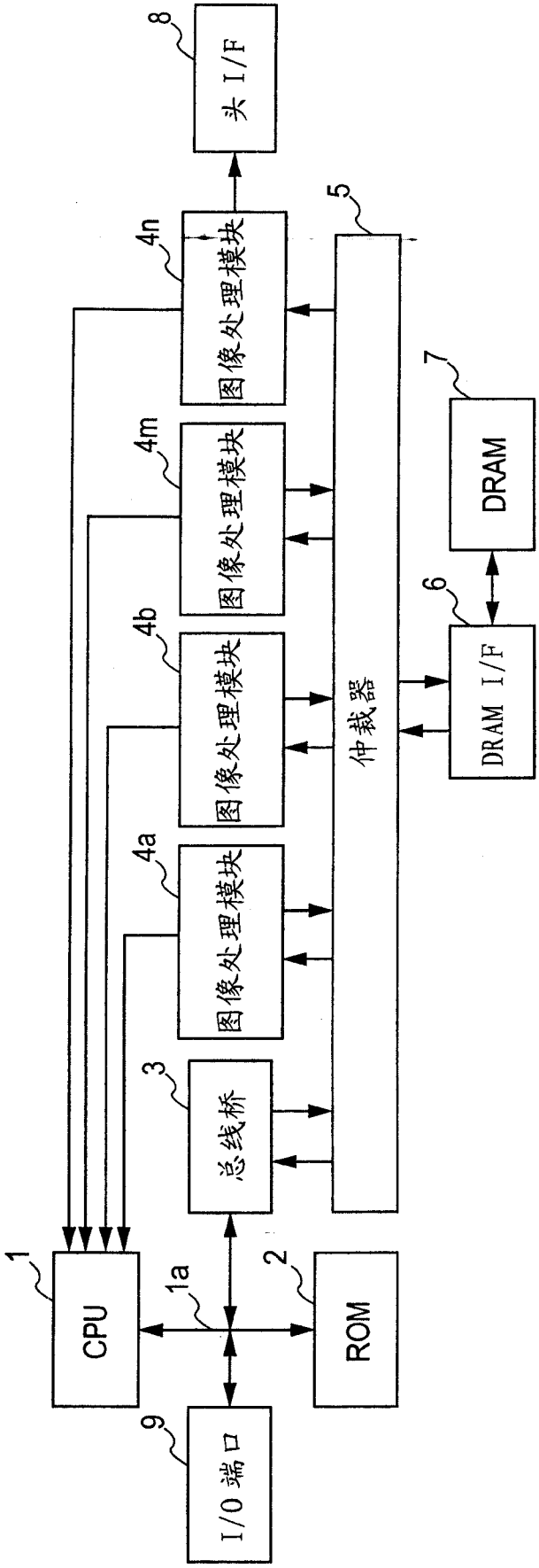

CPU
1
1a
ROM
2
I/O 端口
9
总线桥
3
图像处理模块
4a
图像处理模块
4b
图像处理模块
4m
图像处理模块
4n
仲裁器
5
DRAM I/F
6
DRAM
7
头 I/F
8

图 1

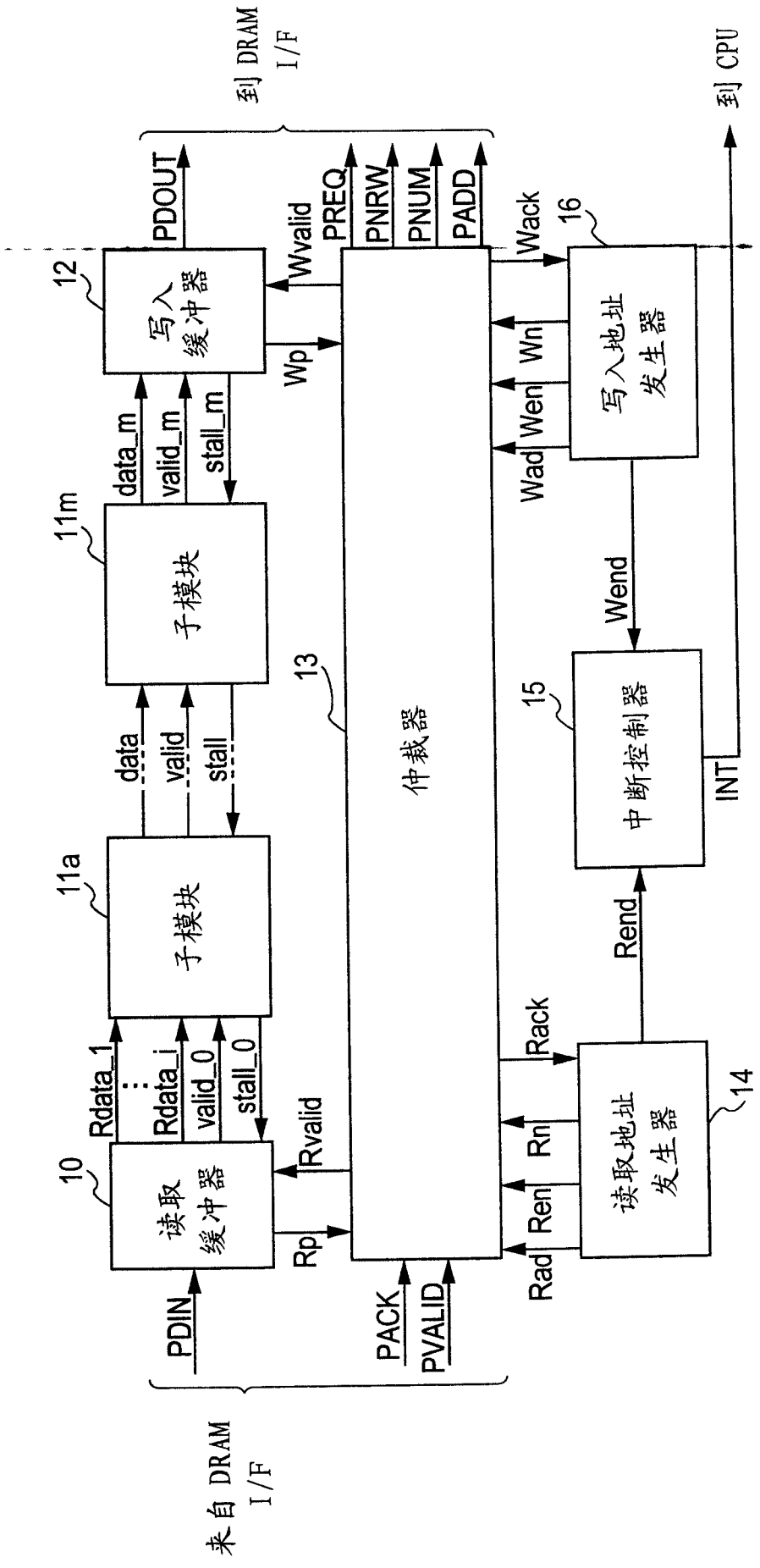

来自 DRAM I/F
到 DRAM I/F
到 CPU
PDIN
PDOUT
PACK
PVALID
Rdata_1
Rdata_i
valid_0
stall_0
Rvalid
Rp
Rack
Rn
Ren
Rad
Rend
data
valid
stall
data_m
valid_m
stall_m
Wp
Wvalid
PREQ
PNRW
PNUM
PADD
Wack
Wn
Wen
Wad
Wend
INT
10
11a
11m
12
13
14
15
16
读取缓冲器
子模块
子模块
写入缓冲器
仲裁器
读取地址发生器
中断控制器
写入地址发生器

图 2

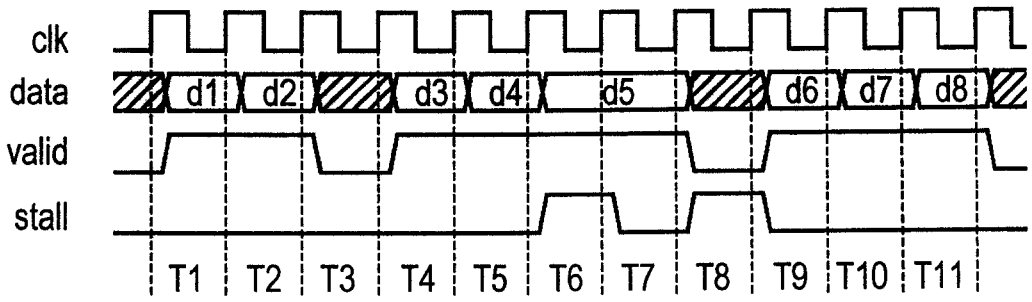

clk
data
valid
stall
d1
d2
d3
d4
d5
d6
d7
d8
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11

图 3

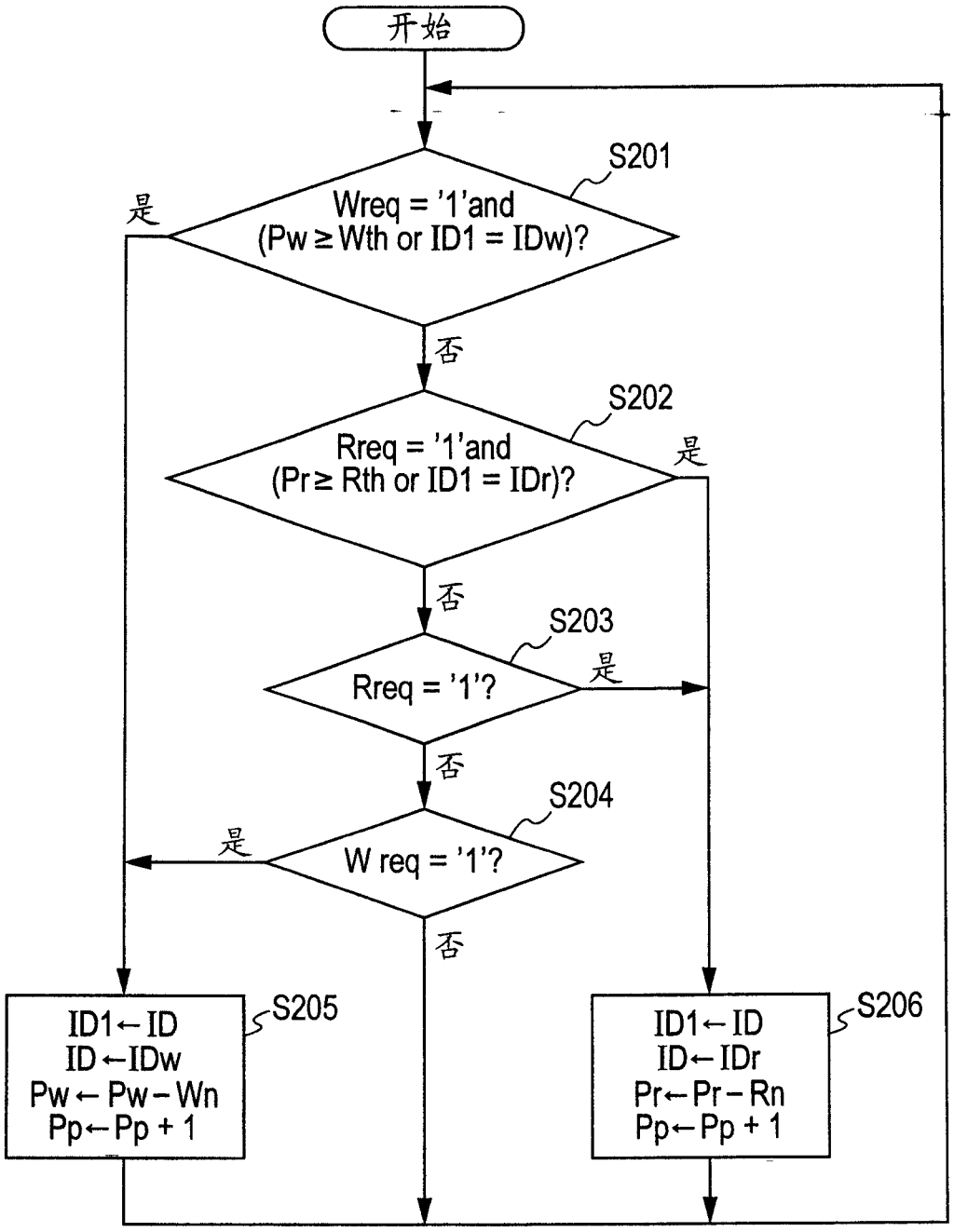

开始
S201
Wreq = '1'and
(Pw ≥ Wth or ID1 = IDw)?
是
否
S202
Rreq = '1'and
(Pr ≥ Rth or ID1 = IDr)?
是
否
S203
Rreq = '1'?
是
否
S204
W req = '1'?
是
否
ID1←ID
ID←IDw
Pw←Pw−Wn
Pp←Pp + 1
S205
ID1←ID
ID←IDr
Pr←Pr−Rn
Pp←Pp + 1
S206

图 4

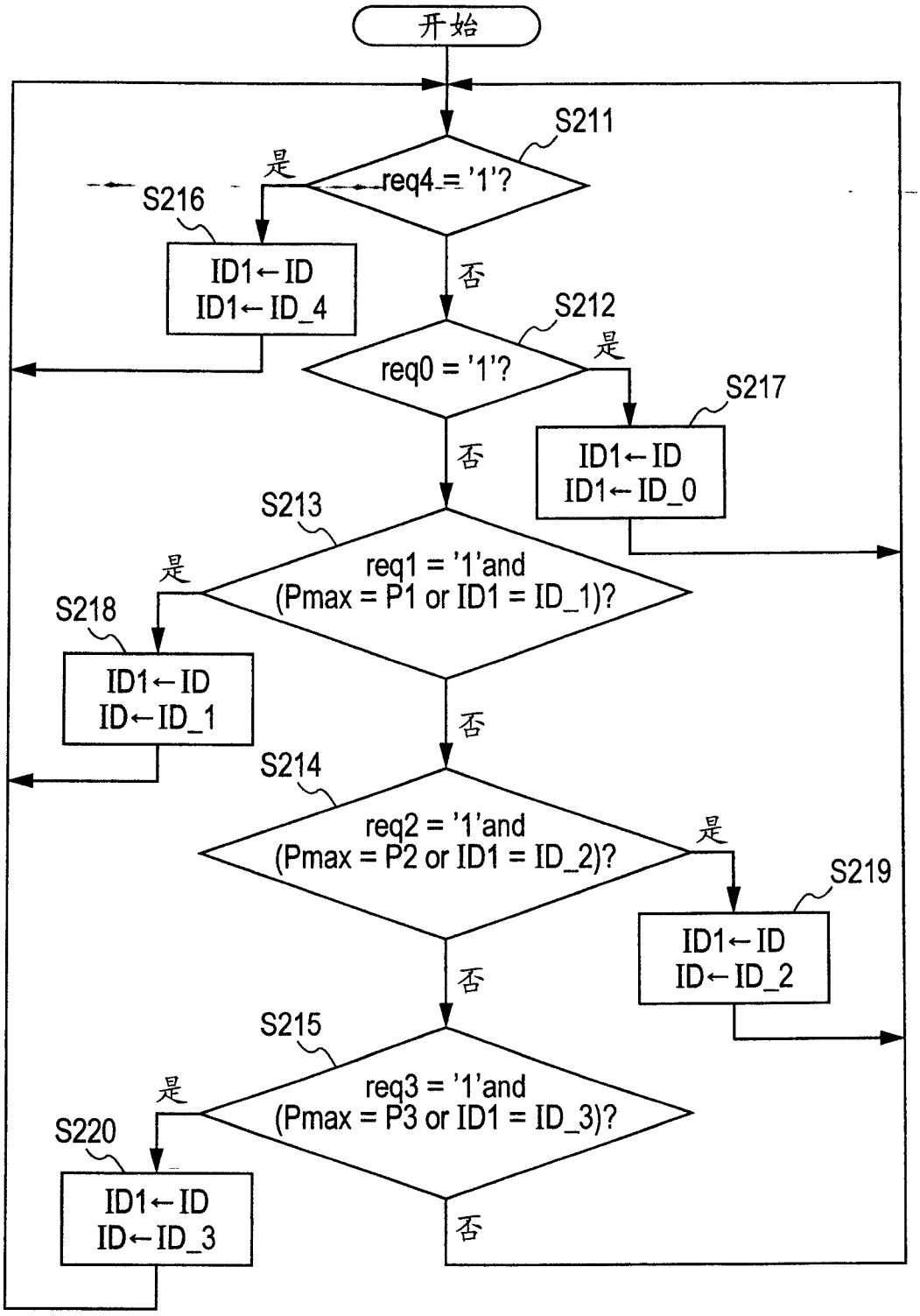

开始
S211
req4 = '1'?
是
S216
ID1←ID
ID1←ID_4
否
S212
req0 = '1'?
是
S217
ID1←ID
ID1←ID_0
否
S213
req1 = '1'and
(Pmax = P1 or ID1 = ID_1)?
是
S218
ID1←ID
ID←ID_1
否
S214
req2 = '1'and
(Pmax = P2 or ID1 = ID_2)?
是
S219
ID1←ID
ID←ID_2
否
S215
req3 = '1'and
(Pmax = P3 or ID1 = ID_3)?
是
S220
ID1←ID
ID←ID_3
否

图 5

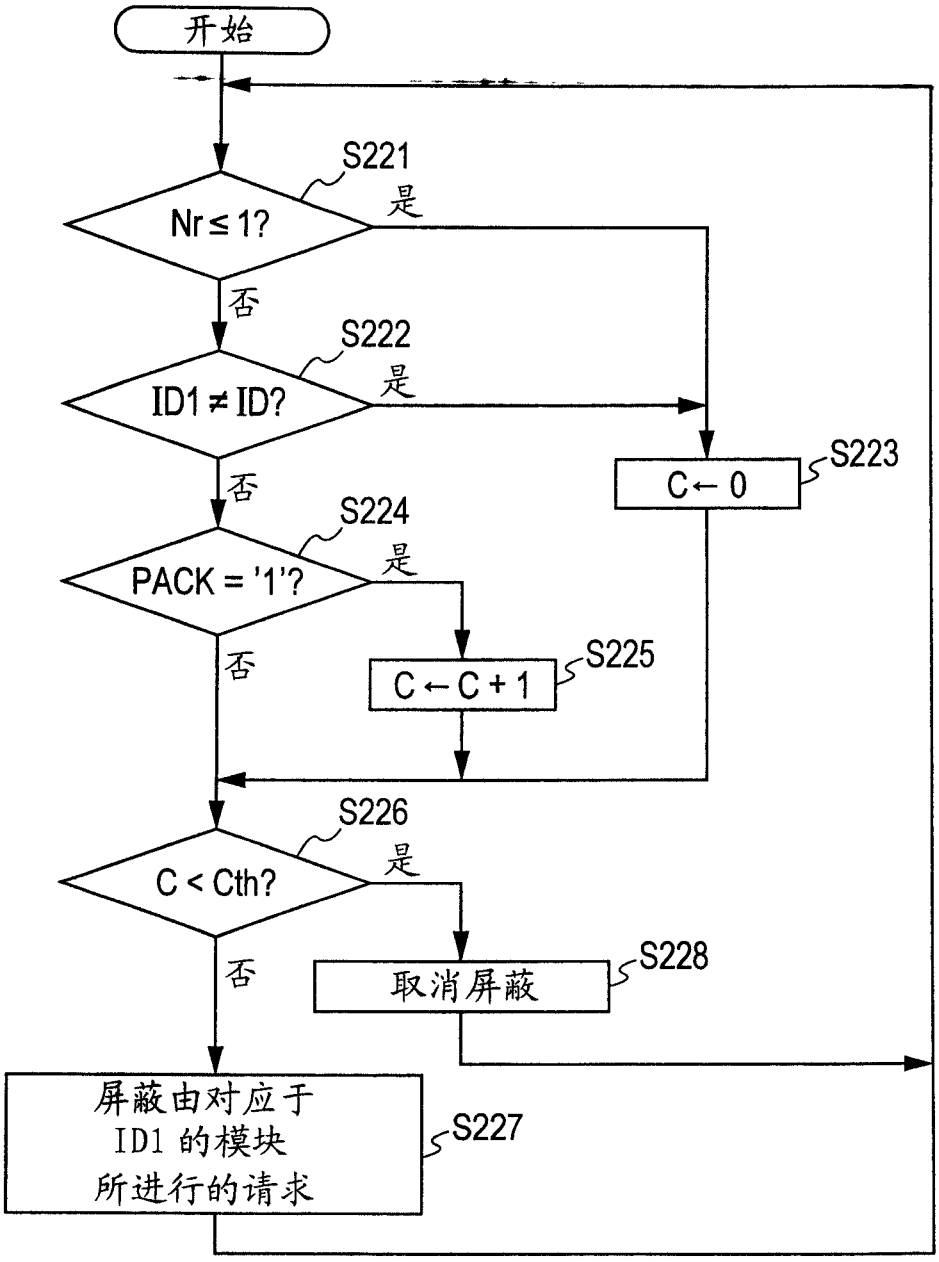

开始
S221
Nr ≤ 1?
是
否
S222
ID1 ≠ ID?
是
否
C ← 0
S223
S224
PACK = '1'?
是
否
C ← C + 1
S225
S226
C < Cth?
是
否
取消屏蔽
S228
屏蔽由对应于
ID1 的模块
所进行的请求
S227

图 6

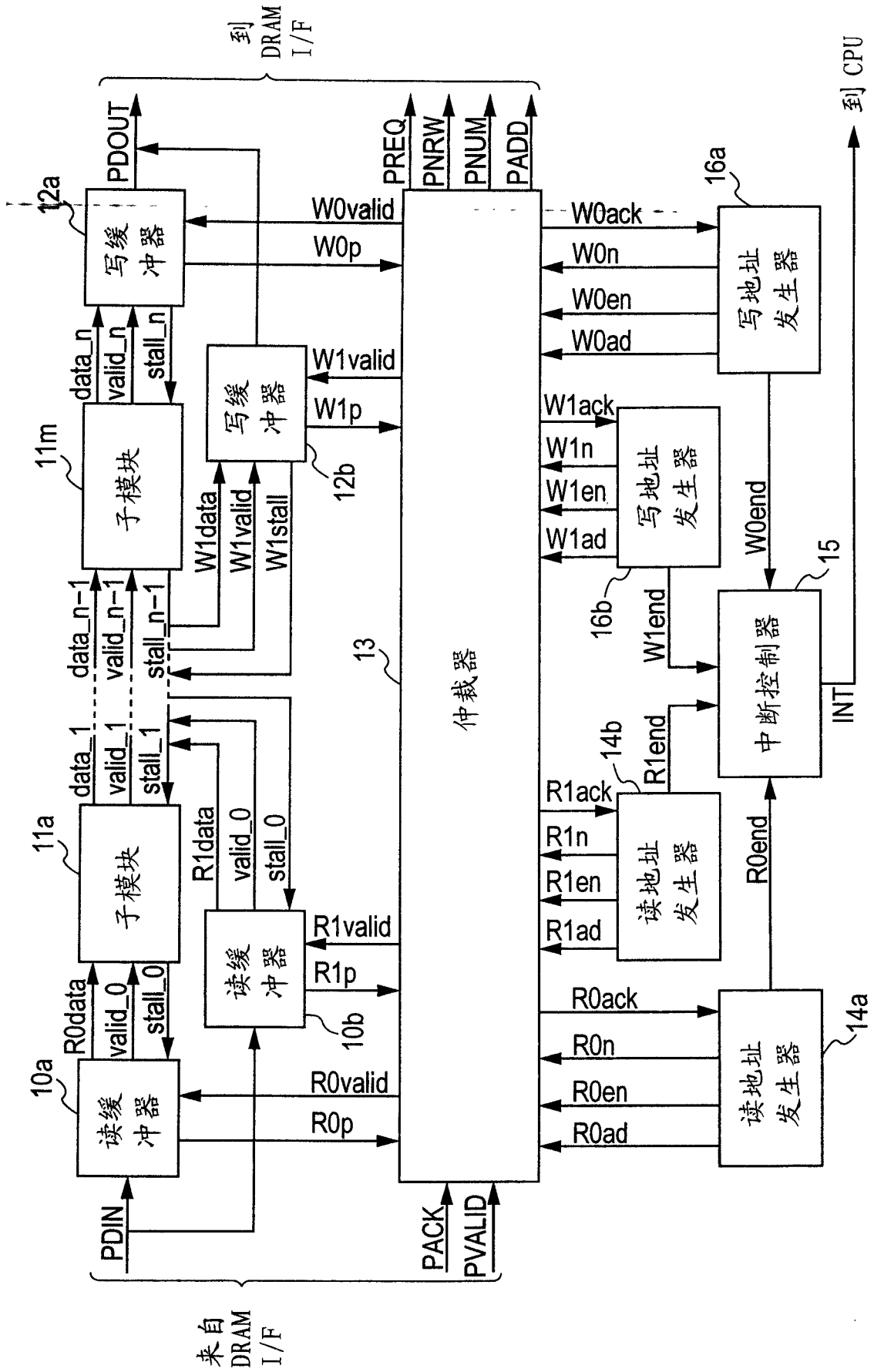

到 DRAM I/F
PDOUT
PREQ
PNRW
PNUM
PADD
12a
写缓冲器
W0valid
W0p
W0ack
W0n
W0en
W0ad
16a
写地址发生器
data_n
valid_n
stall_n
W1valid
W1p
W1ack
W1n
W1en
W1ad
11m
子模块
写缓冲器
12b
W1data
W1valid
W1stall
写地址发生器
W0end
data_n-1
valid_n-1
stall_n-1
13
仲裁器
16b
W1end
15
data_1
valid_1
stall_1
中断控制器
INT
到 CPU
R1data
valid_0
stall_0
R1valid
R1p
R1ack
R1n
R1en
R1ad
14b
R1end
11a
子模块
读缓冲器
读地址发生器
R0end
R0data
valid_0
stall_0
10b
R0valid
R0p
R0ack
R0n
R0en
R0ad
10a
读缓冲器
读地址发生器
14a
PDIN
PACK
PVALID
来自 DRAM I/F

图 7

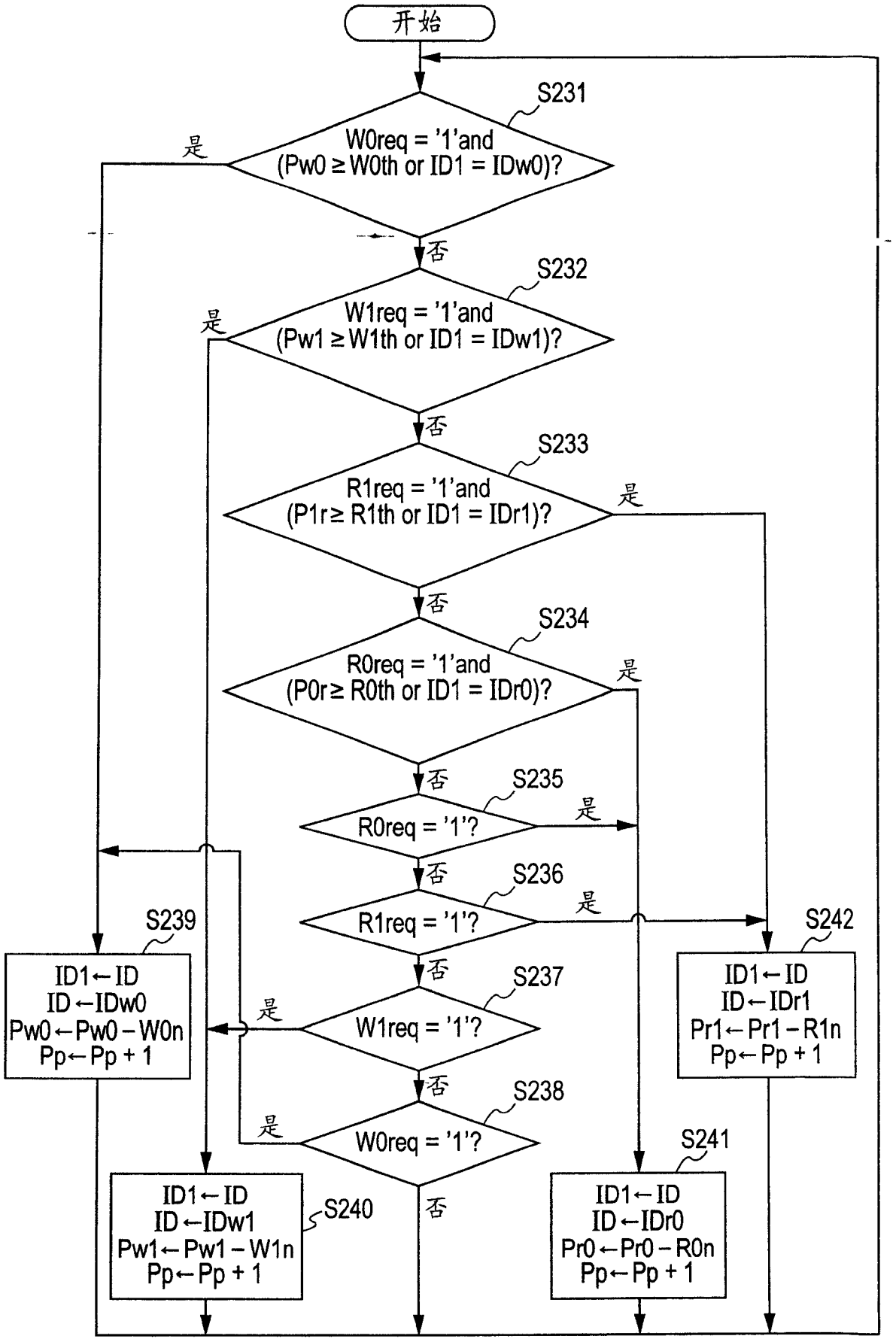

开始
S231
W0req = '1'and
(Pw0 ≥ W0th or ID1 = IDw0)?
是
否
S232
W1req = '1'and
(Pw1 ≥ W1th or ID1 = IDw1)?
是
否
S233
R1req = '1'and
(P1r≥ R1th or ID1 = IDr1)?
是
否
S234
R0req = '1'and
(P0r≥ R0th or ID1 = IDr0)?
是
否
S235
R0req = '1'?
是
否
S236
R1req = '1'?
是
否
S237
W1req = '1'?
是
否
S238
W0req = '1'?
是
否
S239
ID1←ID
ID←IDw0
Pw0←Pw0−W0n
Pp←Pp + 1
S240
ID1←ID
ID←IDw1
Pw1←Pw1−W1n
Pp←Pp + 1
S241
ID1←ID
ID←IDr0
Pr0←Pr0−R0n
Pp←Pp + 1
S242
ID1←ID
ID←IDr1
Pr1←Pr1−R1n
Pp←Pp + 1

图 8