[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/445

[12] 发明专利说明书

[21] ZL 专利号 95193125.3

[43] 授权公告日 2003 年 4 月 23 日

[11] 授权公告号 CN 1106609C

[22] 申请日 1995.4.18 [21] 申请号 95193125.3
[30] 优先权
[32] 1994. 5.18 [33] US [31] 08/245,362
[86] 国际申请 PCT/US95/04865 1995.4.18
[87] 国际公布 WO95/31777 英 1995.11.23
[85] 进入国家阶段日期 1996.11.18
[71] 专利权人 英特尔公司
地址 美国加利福尼亚州
[72] 发明人 S·S·梅力迪思
审查员 杨 蕊

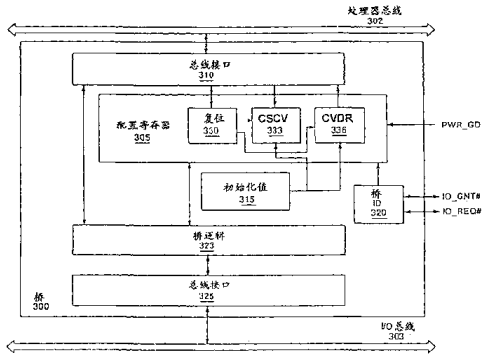
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 王 勇 邹光新

权利要求书 4 页 说明书 14 页 附图 5 页

[54] 发明名称 在一个计算机系统中配置多个媒体的方法和设备

[57] 摘要

在计算机系统中配置多个媒体的设备(300)，包括存储一组配置值的第一存储设备(315)和从总线(302)上捕获该组配置值的第二存储设备(333)。这些配置值被驱动到总线(302)上，并由在本设备(300)内的接口逻辑(310)从总线(302)检索。当本设备(300)中的一个寄存器(336)被激活时，接口逻辑(310)驱动配置值到总线(302)。在系统加电时第一存储设备(315)取第一组值为缺省值。然后这些值被驱动到总线(302)上，并由总线(302)上的每一媒体(203—206)检索。该第一组值提供系统为访问和执行一定的初始化程序的必需的参数。这些初始化程序可以修改存储在第一存储设备(315)中的配置值。在初始化程序修改第一存储设备(315)之后，它们激活本设备(300)中的一个寄存器(336)。该寄存器(336)引起接口逻辑(310)驱动在第一存储设备(315)中存储的值到系统总线(302)上。总线(302)上的每一个媒体(203—206)，包括本设备(300)检索这些值。本设备(300)存储这些值到第二存储设备(333)中并根据存储在第二存储设备(333)中的值继续运行。



ISSN1008-4274

1. 用多个配置值配置在一个计算机系统中的多个媒体的设备，其中该计算机系统具有连接至多个媒体的总线，并且其中所述设备包括：

储存所述多个配置值的第一存储装置；

接口逻辑，用于驱动所述多个配置值从所述第一存储装置到所述总线，由多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量，和用于控制在所述设备操作过程中对所述第一存储装置的修改；

存储所述多个配置值的第二存储装置，其中该设备利用存储在所述第二存储装置内的所述多个配置值来为该设备定义多个操作参量；和

第一寄存器，用于判定所述接口逻辑何时驱动所述多个配置值从所述第一存储装置到所述总线，和用于判定何时所述第二存储装置存储所述多个配置值。

2. 如权利要求 1 所述的设备，其特征在于所述接口逻辑也用于在驱动所述多个配置值到所述总线时，将所述多个配置值存储到所述第二存储装置中。

3. 如权利要求 1 所述的设备，其特征在于所述第一存储装置和所述第二存储装置分别包括一个第二寄存器和一个第三寄存器。

4. 如权利要求 1 所述的设备，其中所述接口逻辑在对所述第一寄存器写操作时驱动所述多个配置值到所述总线。

5. 如权利要求 1 所述的设备，其特征在于所述多个媒体的每一个都是一个处理器。

6. 如权利要求 1 所述的设备，其特征在于所述接口逻辑也用于向所述多个媒体发出一个复位信号，所述复位信号向所述多个媒体的每一个指出何时检索和存储所述多个配置值。

7. 用多个配置值同时配置在计算机系统中的多个媒体的方法，其中

该计算机系统具有连接至多个媒体的总线，所述方法包括以下步骤：

(a) 驱动第二多个配置值到所述总线；

(b) 根据第二多个配置值，访问第一存储装置判定所述第一多个配置值，其中所述第一多个配置值不同于第二多个配置值；

(c) 将所述第一多个配置值存储到第二存储装置；

(d) 将所述第一多个配置值从第二存储装置驱动到所述总线；

(e) 由所述多个媒体的每一个同时检索所述第一多个配置值；和

(f) 由所述多个媒体的每一个同时存储所述第一配置值以便同时为所述的多个媒体的每一个判定不同的操作参量。

8. 如权利要求 7 所述的方法，其特征在于所述访问步骤(b)包括根据所述第二多个配置值判定访问所述第一存储设备的一种方法的步骤。

9. 如权利要求 7 所述的方法，其特征在于所述驱动步骤(d)包括启动一个寄存器使得所述第一配置值被驱动到所述总线上去的步骤。

10. 如权利要求 9 所述的方法，其特征在于所述启动一个寄存器的步骤包括写入所述寄存器的步骤。

11. 如权利要求 7 所述的方法，其特征在于所述检索步骤(e)包括所述多个媒体的每一个在同一个时钟周期内检索所述第一多个配置值。

12. 如权利要求 7 所述的方法，其特征在于还包括将一个复位信号与所述第一多个配置值一起驱动到所述总线，其中所述多个媒体的每一个检索和存储对应于该复位信号的所述第一多个配置值。

13. 一个计算机系统包括：

第一总线；

第二总线；

连接到所述第一总线的第一多个媒体；

连接到所述第二总线的第二多个媒体；和

既是第一多个媒体中的一个也是第二多个媒体中的一个的第一媒体，其中所述第一媒体包括：

存储多个配置值的第一存储装置，

接口逻辑，用于驱动所述多个配置值从所述第一存储装置到所述第二总线，由所述第二多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量，和用于控制在所述第一媒体操作过程中对所述第一存储装置的修改；

存储所述多个配置值的第二存储装置，其中所述第一媒体利用存储在所述第二存储装置内的所述多个配置值来为所述第一媒体定义多个操作参量；和

第一寄存器，用于判定所述接口逻辑何时驱动所述多个配置值从所述第一存储装置到所述第二总线，和用于判定何时所述第二存储装置存储所述多个配置值。

14. 如权利要求 13 所述的计算机系统，其特征在于所述第二多个媒体的一个第二媒体包括一个存储来自所述总线的所述多个配置值的第三存储装置。

15. 如权利要求 13 所述的计算机系统，其特征在于所述第二多个媒体的一个第二媒体包括用于根据所述第一多个配置值从所述第一多个媒体中的一个第三媒体检索一个第二多个配置值的逻辑。

16. 如权利要求 13 所述的计算机系统，其特征在于所述第一媒体为一个总线桥，所述第二多个媒体中的每一个都是一个处理器。

17. 如权利要求 13 所述的计算机系统，其特征在于所述接口逻辑也用于驱动一个复位信号到所述第二总线，所述复位信号向所述第二多个媒体的每一个指出何时检索和存储所述多个配置值。

18. 用多个配置值配置在一个计算机系统中的多个媒体的设备，其中该计算机系统具有连接至多个媒体的总线，并且所述设备包括：

储存所述多个配置值的第一存储装置；

一个驱动装置，用于驱动所述多个配置值从所述第一存储装置到所述总线，由多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量；

一个控制装置，用于控制在所述设备操作过程中对所述第一存储装置的修改；

第二存储装置，用于在所述驱动装置将所述多个配置值驱动到所述总线之后存储所述多个配置值，其中该设备利用存储在所述第二存储装置内的所述多个配置值来为该设备定义多个操作参量；和

一个判定装置，用于判定所述接口逻辑何时驱动所述多个配置值到所述总线，和何时所述第二存储装置存储所述多个配置值。

19.如权利要求 18 所述的设备，其特征在于所述驱动装置还包括将所述多个配置值存储在所述第二存储装置内的装置。

20. 如权利要求 18 所述的设备，其特征在于所述驱动装置在所述判定装置被写操作时驱动所述多个配置值到所述总线上。

在一个计算机系统中配置多个媒体的方法和设备

发明背景

发明领域

本发明涉及计算机系统。更为特别地说本发明是关于在一个计算机系统内无跳线配置多个媒体(agent)。

背景

现代计算机系统常常包括连接到系统总线上的多个媒体。这些媒体可能包括像微处理器、存储器设备、输入／输出（Ｉ／Ｏ）设备等这样的设备。为使计算机系统适当运行，这些媒体必须一起作用来执行它们各自的任务。

保证计算机系统中的媒体一起作用的一种方式是建立一组配置值。这些配置值通常包括所有媒体必须一致认同的项目，像错误检查协议等。因此，当这些配置值对系统中的每一媒体都相同时，系统中的媒体可适当地一起作用。

除了需要媒体之间对配置值一致认同的要求外，所有媒体应该对一定的配置值在同一时间达到一致。例如，一个通常的配置值定义在系统总线上传输的错误检查协议；如果对该协议未达到一致，则一个目的媒体将不能向源媒体指示发生了一个错误。另外，每一媒体应该在同一时间对错误检查协议达到一致；如果未在同一时间达到一致，则存在一段时期，在该时期内媒体使用不同的错误检查协议。在该时间期间内媒体不能适当地彼此指示出错误。

建立这些配置值（也就是说，所有媒体必须在同一时间达成一致的那些配置值）的通常的方法是为每一媒体设立一个跳线。使用跳线时，系统中的每一媒体有一组系统用户可够到的针，用跳线连接适当的针的组合，系统用户能够为该媒体设定合适的配置值。通过把跳线适当地放在每一媒体上，系统用户能够为每一媒体设置同样的配置值。于是，当系统加电时，每一媒体具有合适的配置值，且它们都在同一时间设定（也就是说，在系统加电时）。

为改变一个特定媒体的配置值，用户必须改变跳线的设置。于是，为改变所有媒体在总线上的配置值，一般需要对每一媒体的每一个跳线进行手工操作。另外，一般跳线的改变必须在系统关断后进行。虽然在系统运行时可能手工改变跳线，但是这样的行动并不能同时改变所有的配置值。另外，在

系统运行时改变跳线可能产生不可预测的系统行为。

另外，跳线的使用在两方面增加了系统成本。第一，针和跳线的存在是额外的材料和制造成本。第二，在制造时要花时间配置系统（也就是说放置跳线），这也增加了成本。虽然第二种成本在终端用户而非制造商配置系统时可能不存在，但需要用户进行这样的操作增加了用户的不便。

随着现代计算机系统的性能持续增长，计算机系统的使用也持续扩展。因此，提供一种容易改变以适应广泛应用和多种用途的计算机系统是有益的。然而，为改变用途，需要改变存储在系统中的配置值。因此，提供一种系统，其允许系统用户容易地修改它的配置值以适应不同用途是有益的。本发明提供这样的解决方案。

另外，在一些领域中，计算机系统的使用需要在系统运行时改变。在这样的情况下，可能需要改变配置值。因此，提供一种系统，它允许在系统运行时可以迅速而方便地改变它的配置值是有益的。本发明提供这样的解决方案。

发明概述

本发明包括在一个计算机系统中配置多个媒体的一种方法和设备。

本发明提供了用多个配置值配置在一个计算机系统中的多个媒体的设备，其中该计算机系统具有连接至多个媒体的总线，并且其中所述设备包括：

储存所述多个配置值的第一存储装置；

接口逻辑，用于驱动所述多个配置值从所述第一存储装置到所述总线，由多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量，和用于控制在所述设备操作过程中对所述第一存储装置的修改；

存储所述多个配置值的第二存储装置，其中该设备利用存储在所述第二存储装置内的所述多个配置值来为该设备定义多个操作参量；和

第一寄存器，用于判定所述接口逻辑何时驱动所述多个配置值从所述第一存储装置到所述总线，和用于判定何时所述第二存储装置存储所述多个配置值。

本发明提供了用多个配置值同时配置在计算机系统中的多个媒体的方法，其中该计算机系统具有连接至多个媒体的总线，所述方法包括以下步骤：

(a)驱动第二多个配置值到所述总线；

(b)根据第二多个配置值，访问第一存储装置判定所述第一多个配

置值，其中所述第一多个配置值不同于第二多个配置值；

(c)将所述第一多个配置值存储到第二存储装置；

(d)将所述第一多个配置值从第二存储装置驱动到所述总线；

(e)由所述多个媒体的每一个同时检索所述第一多个配置值；和

(f)由所述多个媒体的每一个同时存储所述第一配置值以便同时为所述的多个媒体的每一个判定不同的操作参量。

本发明还提供了一个计算机系统，包括：

第一总线；

第二总线；

连接到所述第一总线的第一多个媒体；

连接到所述第二总线的第二多个媒体；和

既是第一多个媒体中的一个也是第二多个媒体中的一个的第一媒体，其中所述第一媒体包括：

存储多个配置值的第一存储装置，

接口逻辑，用于驱动所述多个配置值从所述第一存储装置到所述第二总线，由所述第二多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量，和用于控制在所述第一媒体操作过程中对所述第一存储装置的修改；

存储所述多个配置值的第二存储装置，其中所述第一媒体利用存储在所述第二存储装置内的所述多个配置值来为所述第一媒体定义多个操作参量；和

第一寄存器，用于判定所述接口逻辑何时驱动所述多个配置值从所述第一存储装置到所述第二总线，和用于判定何时所述第二存储装置存储所述多个配置值。

本发明还提供了用多个配置值配置在一个计算机系统中的多个媒体的设备，其中该计算机系统具有连接至多个媒体的总线，并且所述设备包括：

储存所述多个配置值的第一存储装置；

一个驱动装置，用于驱动所述多个配置值从所述第一存储装置到所述总线，由多个媒体的每一个进行检索和存储，以便在所述多个媒体的每一个中配置出多个不同的操作参量；

一个控制装置，用于控制在所述设备操作过程中对所述第一存储装置的修改；

第二存储装置，用于在所述驱动装置将所述多个配置值驱动到所述

总线之后存储所述多个配置值，其中该设备利用存储在所述第二存储装置内的所述多个配置值来为该设备定义多个操作参量；和

一个判定装置，用于判定所述接口逻辑何时驱动所述多个配置值到所述总线，和何时所述第二存储装置存储所述多个配置值。

在一个实施例中，计算机系统有一条总线和一个设备，该设备具有第一存储设备用以存储一组配置值。配置值由在该设备中的接口逻辑驱向总线。该设备还包括一个第二存储设备，用以从总线上捕获该组配置值。当在该装置中的一个寄存器被激活时，接口逻辑驱动配置值到总线。

在系统加电时，第一存储设备取第一组值为缺省值。然后这些值被驱向总线并且被总线上的每一媒体捕获。该第一组值给系统提供必需的参量以访问和执行某些初始化程序。这些初始化程序可以修改存储在第一存储设备中的配置值，从而以希望的配置值设置该第一存储设备供系统使用。

在初始化程序修改第一存储设备后，激活在本发明的设备中的一个寄存器。该寄存器引起接口逻辑驱动存储在第一存储设备中的值到系统总线。总线上的每一媒体，包括本发明的设备，捕获这些值。本发明的设备把这些值存储在第二存储设备中，并按照存储在第二存储设备中的值继续运行。

在系统运行时可以用软件对系统的配置值进行后继的修改。这些修改是通过对第一存储设备中的值修改而实现的。一旦完成这些修改，该软件激活该设备内的寄存器，从而引起总线上的所有媒体从总线上捕获新的配置值。

附图的简要说明

本发明以示例方式说明而不局限于附图中的图形，图中相同的数字表示相似的元件，其中：

图 1 是表示本发明的示例多处理器计算机系统的总图；

图 2 是表示本发明的示例总线群集系统的框图；

图 3 是表示本发明的一个实施例的设备的框图；

图 4 表示驱动本发明的一个实施例中的初始化配置值的定时图；

图 5 是表示在本发明的一个实施例中更新媒体配置值的步骤的流程图。

在下面的详细说明中，叙述了大量特别的细节，以便提供对本发明的彻底理解。然而对熟悉本技术领域的人来说应该理解，本发明可以不要这些特别的细节而实现。在其它的例子中，公知的方法、过程、元件和电路未详细叙述，以便不致搞混本发明的方面。

下面详细说明的一些部分是根据在计算机存储器内对数据位操作的算法和符号描述介绍的。这些算法说明和叙述由在数据处理技术领域熟悉的人员

使用，它是最有效地对该技术领域其他熟练人员表达他们工作实质的方法。在这里，通常一个算法被表达为引向一个希望的结果的一系列自身一致的步骤。这些步骤是需要手工操纵物理量的那一些。虽然不一定必须，但通常这些数量采取电子或磁信号形式，能够被存储、传输、结合、比较以及其它操作。已经证明，有时特别是在为公共使用时称这些信号为位、值、元素、符号、字符、术语、数字或其它是方便的。

然而应该记住，所有这些和类似的术语是与适当的物理量相关联的，并且仅仅是应用于这些数量方便的标签。除非特别指出，否则如同下面的讨论中明显看出的，可以理解，在整个本发明当中使用像“处理”或“计算”或“运算”或“确定”或“显示”之类的术语指的是计算机系统或类似的电子计算设备的动作和处理，这些设备将描述为在该计算机系统的寄存器和存储器中的物理（电子）量的数据操作和转换为其它相似地描述为在该计算机系统的存储器或寄存器内或其它这样的信息存储、传输或显示的设备内的数据。

图１是表示本发明的一个示例多处理器计算机系统的总图。该计算机系统通常包括一个处理器－存储器总线或其它通信设备１０１，以便在一个或多个处理器１０２、１０３、１０４和１０５之间交换信息。处理器－存储器总线１０１包括地址、数据和控制总线。处理器１０２到１０５可以包括一个小的、极快的内部高速缓冲存储器，一般称为本地（Ｌ１）高速缓冲存储器，用于临时存储数据和指令到芯片。另外，一个较大的、较慢的第二（Ｌ２）高速缓冲存储器１０６可以连接到一个处理器，像处理器１０５，以临时存储数据和指令供处理器１０５使用。一种方式是，本发明包括英特尔（Ｉｎｔｅｌ）结构微处理器，如处理器１０２到１０５；然而本发明也可以使用任何类型的微处理器结构，或任何数字信号处理器的主机。

处理器１０２、１０３和１０４包括一个并行处理器，例如一个相似于或相同于处理器１０５的处理器。可选的另外的方案为，处理器１０２、１０３和１０４包括一个协处理器，例如一个数字信号处理器。另外，处理器１０２到１０５可以包括不同类型的处理器。

处理器－存储器总线１０１提供对存储器和输入／输出（Ｉ／Ｏ）子系统的系统存取。一个存储器控制器１２２与处理器－存储器总线 101 连接，以控制对一个随机存取存储器（ＲＡＭ）或其它动态存储设备１２１（通常称为主存储器）的存取，以便为处理器１０２到１０５存储信息和指令。用以存储信息和指令的一个海量数据存储设备１２５，诸如一个磁盘和磁盘驱动器，和用以显示信息给计算机用户的一个显示设备１２３，诸如阴极射线

管（ＣＲＴ）、液晶显示器（ＬＣＤ）等可以连接到处理器存储器总线１０１上。

一个输入／输出（Ｉ／Ｏ）桥１２４可以连接到处理器－存储器总线１０１和系统Ｉ／Ｏ总线１３１上，为在处理器－存储器总线１０１或者Ｉ／Ｏ总线１３１上的设备提供通信路径，以便在其它总线上的设备之间存取和传输数据。本质上，桥１２４是系统Ｉ／Ｏ总线１３１和处理器－存储器总线１０１之间的一个内部接口。

Ｉ／Ｏ总线１３１在计算机系统的外围设备之间交换信息。可以连接到系统总线１３１上的设备包括显示设备，诸如阴极射线管、液晶显示器等；字母数字输入设备１３３，包括字母数字和其它键等，以便把信息和命令选择传输给计算机系统中的其它设备（例如处理器１０２）和用于控制光标的运动光标控制设备。此外，提供计算机图像的视觉描述的硬拷贝设备１３５，像绘图仪或打印机，和存储信息和指令的海量存储设备１３６，像磁盘和磁盘驱动器也可以连接到系统总线１３１上。

另外，一个Ｉ／Ｏ总线１４２可以通过桥１４０连接到处理器－存储器总线１０１上。Ｉ／Ｏ总线１４２可以连接到另外的外围设备上，像连接到系统Ｉ／Ｏ总线１３１的设备１３２到１３６。

在本发明的一个实施例中Ｉ／Ｏ总线１４６通过桥１４５连接到系统Ｉ／Ｏ总线１３１上。Ｉ／Ｏ总线可以连接到另外的外围设备上，例如连接到系统Ｉ／Ｏ总线１３１上的设备１３２到１３６。一种方式是，Ｉ／Ｏ总线１４６以和系统Ｉ／Ｏ总线１３１（例如ＰＣＩ）不同的标准（例如ＥＩＳＡ）运行。

在某些装置中，也许不需要提供显示设备来显示信息。本发明的某些装置可以包括另外的处理器或其它的部件。另外，本发明的某些装置也许不需要包括所有上述部件。例如，处理器１０２到１０４、显示设备１２３、Ｉ／Ｏ总线１４２、或海量存储设备１２５可以不连接到处理器－存储器总线１０１。另外，如图所示连接到系统Ｉ／Ｏ总线１３１上的一些外围设备可以连接到处理器存储器总线１０１上；另外，在一些装置中，仅有单一总线存在，处理器１０２到１０５、存储器控制器１２２、和外围设备１３２到１３６都连接到该单一总线上。

图２是表示本发明的一个示例总线群集系统的框图。本发明可以应用于具有一个或多个处理器群的多处理器计算机系统。图２表示两个这样的群２０１和２０２。每个这样的群包括一些媒体。例如群２０１由４个媒体２０３－２０６和一个群管理器２０７组成，它可以包括连接到总线２１２上的

另一高速缓冲存储器（未示出）。媒体２０３－２０６可以包括微处理器、协处理器、数字信号处理器等；例如媒体２０３到２０６可以与在图１中所示的处理器１０５相同。群管理器２０７及其高速缓冲存储器在这四个媒体２０３－２０６之间共享。每一群连接到一个存储器－系统总线２０８上，其在一个实施例中是图１的处理器－存储器总线１０１。这些群２０１－２０２通过一个系统接口２０９连接到该计算机系统的各种其它的部件上。系统接口２０９包括一个Ｉ／Ｏ接口２１０作为该计算机系统对外部世界的接口，和一个存储器接口２１１，它提供对像ＤＲＡＭ存储器阵列的主存储器的存取。在一个实施例中，Ｉ／Ｏ接口２１０是图１中的桥124，而存储器接口２１１是图１中的存储器控制器１２２。

在一个实施例中，每一群也包括一个本地存储器控制器和／或一个本地Ｉ／Ｏ桥。例如，群２０１可以包括一个连接到处理器总线２１２上的局部存储器控制器２６５。局部存储器控制器２６５管理对ＲＡＭ或在群２０１内包含的其它局部存储器设备２６６的存取。群２０１也可以包括一个连接到处理器总线２１２上的Ｉ／Ｏ桥２６７。局部Ｉ／Ｏ桥２６７管理对群内Ｉ／Ｏ设备的存取，像显示设备２７０或连接到局部Ｉ／Ｏ总线２６８上的海量存储设备２７１。一种方式为，每一群２０１和２０２包括一个连接到本地Ｉ／Ｏ总线上的本地Ｉ／Ｏ桥，其连接到图１中示出的外围设备１３２到１３６上。

本发明的某些设备也许既不需要也不包括所有上述部件。例如，群２０１或２０２可以包括少于４个媒体。另外，本发明的某些设备可以包括另外的处理器或其它部件。

在本发明的一个实施例中，图１和图２的计算机系统是个人计算机系统。这些系统可以是，例如，桌上型或便携式计算机系统。

图３是表示本发明的一个实施例的设备的框图。图中所示的桥３００包括配置寄存器３０５、总线接口３１０、初始化值３１５、桥ＩＤ３２０、桥逻辑３２３和总线接口３２５。桥３００通过接口３１０连接到处理器总线３０２和通过接口３２５连接到Ｉ／Ｏ总线３０３。总线３０２和总线３０３两者都包括一个地址总线、数据总线、和控制总线，像在该技术领域中熟知的那样。在一个实施例中，桥３００是图１中的桥１２４。在一个可选的另外实施例中，桥３００是图２中的Ｉ／Ｏ接口２１０。

在后面的讨论和例子中，本发明是结合在一个总线桥中说明的。然而，对熟悉本技术领域的人员来说很明显，本发明同样可以应用于连接到计算机系统的总线上的任何其它媒体。另外，在一种操作方式中，本发明在总线上

包括它自己的独立设备。

总线接口３２５以传统的方式在Ｉ／Ｏ总线３０３和桥逻辑３２３之间传输地址、数据和控制信号。相似地，总线接口３１０以传统的方式在处理器总线３０２和桥逻辑３２３之间传输地址、数据和控制信号。桥逻辑３２３是桥３００的内部控制逻辑。桥逻辑３２３以传统方式运行，在处理器总线３０２和Ｉ／Ｏ总线３０３之间传输请求。

总线接口３１０还在处理器总线３０２和配置寄存器３０５之间传输地址、数据和控制信号。在一条总线和接在总线上的一个媒体之间这种信号的传输在计算机技术中已经公知，因此不再说明。

配置寄存器３０５包括三个设备：一个复位寄存器３３０，一个捕获系统配置值（ＣＳＣＶ）寄存器３３３和一个配置值驱动复位（ＣＶＤＲ）寄存器３３６。应该注意，虽然配置寄存器３０５是作为寄存器讨论的，但是也可以使用任何其它传统的存储设备。在本发明的一个实施例中，复位寄存器３３０是一个８位寄存器的一位，而ＣＳＣＶ寄存器３３３和ＣＶＤＲ寄存器３３６都是１６位的寄存器。

ＣＳＣＶ寄存器３３３包含由桥３００定义它自己的操作而使用的一组配置值。例如，桥３００使用的按序队列深度就是由在ＣＳＣＶ寄存器３３３中存储的配置值确定的。在一种方式中，连接在处理器总线３０２上的所有媒体都包含一个ＣＳＣＶ寄存器３３３，或它的等价体，它带有一个总线接口，该接口从总线检索配置值。和在桥３００的场合一样，在其它的媒体中的ＣＳＣＶ寄存器３３３也都由那些媒体用于定义自己的操作。于是，通过改变系统中所有其它媒体的ＣＳＣＶ寄存器 333 的值，也就改变了所有媒体的配置值。

表１表示本发明的一个实施例中存储在ＣＳＣＶ寄存器３３３中的配置值。

表１

位	功能
１５：１４	保留
１３	输出三态许可，该位指示处理器是否允许三态输出。当该位置位时，处理器为三态输出。如果该位未置位，处理器为非三态输出。
１２：１１	高级可编程中断控制器（ＡＰＩＣ）群ＩＤ。这些位为连接在总线上的处理器指示 APIC 群 ID。

10	报告BINIT输入。该位指示是否允许BINIT逻辑。如果该位置位，那么当它测定出一种阻止将来信息可靠交换的总线条件时，媒体就确立一个BINIT#信号 。如果该位未置位，则总线上的所有媒体，除中心媒体外，忽略BINIT#信号的确立。
9	报告BERR输入。该位指示是否允许BERR#报告 。当该位置位时，由测知一个不可恢复错误的媒体确立 一个BERR#信号，并且不引起总线规程扰乱。如果该位未置位，总线上的媒体将不确立BERR#信号。
8	使用AERR#的地址总线错误报告。该位指示是否允 许使用AERR#信号的地址奇偶校验报告。当该位置位时，一个AERR#信号的确立指示一个地址奇偶校验错误。如果该位未置位，则除了中心媒体外，在总线上的所有媒体忽略AERR#信号的确立。
7	按序队列深度1。该位指示在总线上的媒体的按序队列深度。按序队列深度是一个事务处理号码，其可以凸出于管线处理器总线上。当该位置位时，队列深度为1。如果该位未置位，则队列深度为8。
6	1M通电复位矢量。该位指示适当复位矢量，根据该矢量处理器在复位后开始执行。当该位置位时，复位矢量为0FFF0H（1兆－16）。如果该位未置位，则复位矢量为0FFFFFFF0H（4千兆－16）。
5	功能冗余检验（FRC）方式允许。当该位置位，系统在FRC方式运行。如果该位未置位，则系统不在FRC方式下运行。
4：0	保留

CVDR寄存器336在通电时以一组缺省配置值初始编程。这些缺省值是初始化值315。在本发明的一个实施例中，初始化值315是系统在最低水平上运行所必需的最小一组的配置值，通过提供一个最小组的值，计算机系统能够合适地运行，虽然也许不能使其打算的使用运行在最优水平上。然而，一旦建立起一最小组的值，就可以以一组新的参量重新为配置寄存器编程，下面将详细叙述。

在一个实施例中，初始化值315是在系统中预先规定的。也就是说，

系统在通电时以特别的初始化值预先配置。这些初始化值可以以大量传统的任何方式预先确定，例如在系统中使用硬件。

该初始化值315也在通电时输入CSCV寄存器333。因而，CSCV寄存器333和CVDR寄存器336在起始时都含有同一组值（亦即初始化值315）。如同下面将要详细叙述的那样，这些值在系统运行时可以改变。

表2表示本发明的一个实施例的初始化值315和相应的CVDR位及其功能。在位已被置位或未被置位之处，用“1”指示该位置位，用“0”指示该位未置位。

表2

位	功能	初始值
15：14	保留	N／A
13	输出三态允许	0
12：11	APIC群ID	00
10	报告BINIT输入	0
9	报告BERR输入	0
8	使用AERR#的地址总线错误报告	0
7	按序队列深度1	0
6	1M通电复位矢量	0
5	FRC方式允许	0
4：0	保留	N／A

当复位寄存器330被激活时，总线接口310驱动包含在CVDR寄存器336中的值到处理器总线302。在一种方式中，这些值经由总线302的地址总线驱动。在一个实施例中，复位寄存器330由软件写操作而被激活。复位寄存器330被激活，还引起总线接口310在总线302上确立一个PROC_RESET#信号。在一种方式中，PROC_RESET#信号是总线302的控制总线的部分。

PROC_RESET#信号由所有连接在处理器总线上的媒体接收，并引起它们执行“复位”操作。该复位操作包括从总线检索配置值并把这些配置值存储在等价于CSCV寄存器333的配置寄存器中。然后，每一个媒体都使用这些新的值执行所有总线上的后继的事务处理。

除把CVDR寄存器336中的值驱动到总线302以外，总线接口3

10还存储这些值到CSCV寄存器333中。这允许桥300随着接在总线上的所有媒体更新它自己的配置值。

图4表示本发明的一个实施例中驱动初始配置值的一个示例定时图。在该实施例中，I／O总线303运行在著名的PCI标准上。如图3中所示，一个PWR_GD信号输入到CVDR寄存器336。PWR_GD是一个由计算机系统的电源产生的信号。在一种方式中，在电源的所有电压稳定1毫秒后，该电源确立一个PWR_GD信号。如图4中所示，PWR_GD信号异步输入到桥300。

在一个实施例中，桥300通过确立一个PCI_RST#信号复位I／O总线303。在一种方式中，每当测知PWR_GD撤销时则确立一个PCI_RST#信号。PCI_RST#信号确立保留到PWR_GD确立后1毫秒。以这种方式，当PCI_RST#从0过渡到1时，在I／O总线303上接的所有其它媒体将复位自身。

在PWR_GD的上升缘，桥300捕获它的桥ID，其为在图3中的桥ID320。它作为桥ID输入表示在图4中。在一个实施例中，桥300在系统运行时维持同样的桥ID；也就是说，PROC_RESET#信号不改变桥ID。桥ID用于判定在包含多个桥的系统中哪一个桥应该驱动配置值到总线，对此下面将作更详细地叙述。

同样在PWR_GD的上升缘，初始化值315写到CSCV寄存器333和CVDR寄存器336。这提供CVDR寄存器336适当的配置值以驱动总线302，下面将要叙述。

一旦接收到PWR_GD，桥300以PROC_RESET#信号复位处理器总线302。PROC_RESET#在PWR_GD从0过渡到1后和时钟同步确立。于是，如图4所示，PWR_GD在CLK2异步确立，而PROC_RESET#在CLK3异步确立。在本发明的一个实施例中，PROC_RESET#保持确立1毫秒。

在PROC_RESET#确立时，配置值从CVDR寄存器336被驱动到处理器总线302。桥300在判定是“兼容桥”后确立配置值；兼容桥是负责配置系统的桥，其在下面将详细叙述。这些值作为配置值表示在图4中，并由总线302上的每一个媒体在PROC_RESET#信号的上升缘检索。于是，在PROC_RESET#被撤销时，连接在总线302上的所有媒体都包含同样的配置值。

图4表示系统通电时初始配置值的驱动。在系统通电后在一定的预定条件下也可以执行其它的复位。例如，可以由系统用户驱动一个物理复位开关。

在这样的条件下，PROC_RESET#信号在复位开关驱动时确立，而在CVDR寄存器336中的值被驱动到总线，如上面所述。

在一个实施例中，物理复位开关的驱动引起初始化值315加载到CVDR寄存器336中。以另一可选择的方式，初始化值315不加载到CVDR寄存器336中。在这种可选方式中，在复位开关被驱动时存储在CVDR寄存器336中的配置值被驱动到总线302。

在本发明的一个实施例中，系统也可以由写入复位寄存器330而复位。在该实施例中，PROC_RESET#信号和配置值输入像图4所示确立。初始化值315不置换存储在CVDR寄存器336中的值。而不管复位寄存器330被写入时在CVDR寄存器中存储的是什么值，它都被驱动到总线。

于是，可以看到，新配置值可以存储在CVDR寄存器336中并接着在复位期间被驱动到总线。这种复位是由对复位寄存器330写操作而控制的。

如图1 和图2所示，一个计算机系统可以包括多于一个的总线桥。在一个实施例中，每一桥都等同于图3中的桥300。于是，每一桥包含配置寄存器305。为使在该实施例中多个桥适当作用，只允许一个桥去复位处理器总线。是否允许一个桥复位处理器总线是由桥ID320决定的。

在一个实施例中，桥ID320是一两字节值，其在系统加电时建立。桥ID320由IO_GNT#和IO_REQ#信号决定，它们作为图4的桥ID输入被驱动。一种方式为，IO_GNT#和IO_REQ#信号还由桥300在系统加电后用于仲裁处理器总线302。IO_GNT#和IO_REQ#在加电时固定为高或低，因此指示1或0。这可以以大量传统的任何方式实现，例如使用拉上／拉下电阻器或跳线。通过为每一桥固定适当的IO_GNT#和IO_REQ#信号为高或低，可以看到，可以产生四个唯一的桥ID。对熟悉本技术领域的人员来说，可以理解，只要给定合适数目的输入信号，就可以提供任何数目的唯一桥ID。

桥ID用于判定多个桥中的哪一个负责复位总线；负责复位总线的桥称为“兼容桥”。在一个实施例中，每一个桥预先规定带特定桥ID的桥是兼容桥。一种方式为，兼容桥ID是“01”，包括从IO_GNT#信号来的“0”和从IO_REQ#信号来的“1”。

在收到PWR_GD信号时，每一桥开始确立PROC_RESET#信号，如图4所示。因此，当在一个系统中存在多个桥时，每一个桥都确立一个PROC_RESET#信号。然而，每一桥也接收指示它的桥ID的

信号。如果一个桥被判定为兼容桥，则它继续确立PROC_RESET#信号，并且驱动在其CVDR寄存器中的值到总线，如上面所述。然而，如果判定一个桥不是兼容桥，则该桥撤销PROC_RESET#信号。这些非兼容桥不驱动在其CVDR寄存器中的值到总线。相反地，这些桥表现为同连接在总线上的其它媒体一样，亦即这些桥在PROC_RESET#信号的上升缘从总线捕获它们的配置值。

应该注意，在多桥系统中的每一桥的桥ID判定哪一个桥是兼容桥。于是，桥ID只判定哪一个桥负责复位总线，和哪一个桥可以驱动它的CVDR寄存器到总线。哪一个桥执行这些功能并不影响系统通过修改兼容桥的CVDR寄存器来修改所有媒体的配置值的能力。

在一个多桥系统中，兼容桥的CVDR寄存器336可以由系统中的其它媒体（以及兼容桥自身）写入。为做到这一点，一个媒体发布一个发往兼容桥（通过兼容桥的桥ID）的命令。由于每一个桥有唯一的桥ID，而兼容桥的桥ID是预先规定的，则系统中的其它媒体可以合适地找到兼容桥的CVDR寄存器336。

虽然上面给出的是在一个多桥系统中为每一桥建立一个桥ID的特殊细节，但是这些细节并不意味着限制本发明的范围。可以以大量传统的任何方式确定桥ID。

图5表示根据本发明的一个实施例更新配置值的步骤。最初，计算机系统加电，初始配置值驱向总线，步骤510。这一点的实现和上面图4所示一样。

给定初始配置值，处理器总线302上的媒体有必需的参量允许它们访问系统基本输入输出（BIOS），步骤520。一种方式为，BIOS存储在一个连接在总线303上的外部设备中，像海量存储设备。可选的另外方案为，BIOS可以存储在连接在总线302上的设备中，像只读存储器（ROM）或其它连接在总线302上的海量存储设备。

不管BIOS位于何处，初始配置值允许总线上的媒体访问BIOS。BIOS一般包含一组软件程序，其在系统起动时由总线上的处理器执行，如同在该技术领域所熟知的那样。在一个实施例中，这些BIOS程序修改图3中的桥300的CVDR寄存器336的配置值。

在完成修改CVDR寄存器336的配置值后，BIOS程序执行对复位寄存器330的写操作，步骤540。如上所述，这引起总线接口310驱动CVDR寄存器336中的值到总线302，步骤550。总线上的媒体随即检索和存储这些值，并根据这些值运行。

配置值可以在系统运行期间再次改变，如步骤５６０所示。如果未做进一步的修改，则没有另外的配置值驱动到总线。不过，后继的软件程序可以修改ＣＶＤＲ寄存器中的配置值，步骤５７０。如上所述，ＣＶＤＲ寄存器中的值的修改不改变包括桥３００的系统的配置值，直到复位寄存器３３０被写入。在修改ＣＶＤＲ寄存器３３６之后，软件重复步骤５４０和５５０，亦即软件写入复位寄存器３３０，引起ＣＶＤＲ寄存器３３６中新的配置值被驱动到总线。

这种配置值的修改可以在系统运行期间继续，如步骤５６０所示。任何时间，当软件希望修改系统配置值时，可以修改ＣＶＤＲ寄存器，并把其值驱向总线。

因此，可以看到，系统中每一个媒体所用的配置值通过使用复位寄存器３３０、ＣＳＣＶ寄存器３３３和ＣＶＤＲ寄存器３３６同时更新。这些配置值可以在系统运行期间通过预编程合适的寄存器改变任意次；不需跳线或类似设备的物理操作。

应该注意，虽然上面的叙述讨论了修改和驱动一组配置值，但是本发明并不限于这些数值。对熟悉本技术领域的人员来说应该理解本发明同样可以应用于可以在计算机系统中设定的大量的任何值。

鉴于熟悉本技术领域的人员在阅读上述说明后能够领会对本发明的许多改变和修正，所以应该理解，本说明表示和叙述的该特定实施例决不打算限制本发明的范围。因此，对特定实施例的细节的参考并不打算限制权利要求的范围，这些权利要求仅叙述涉及到本发明的基本点的特征。

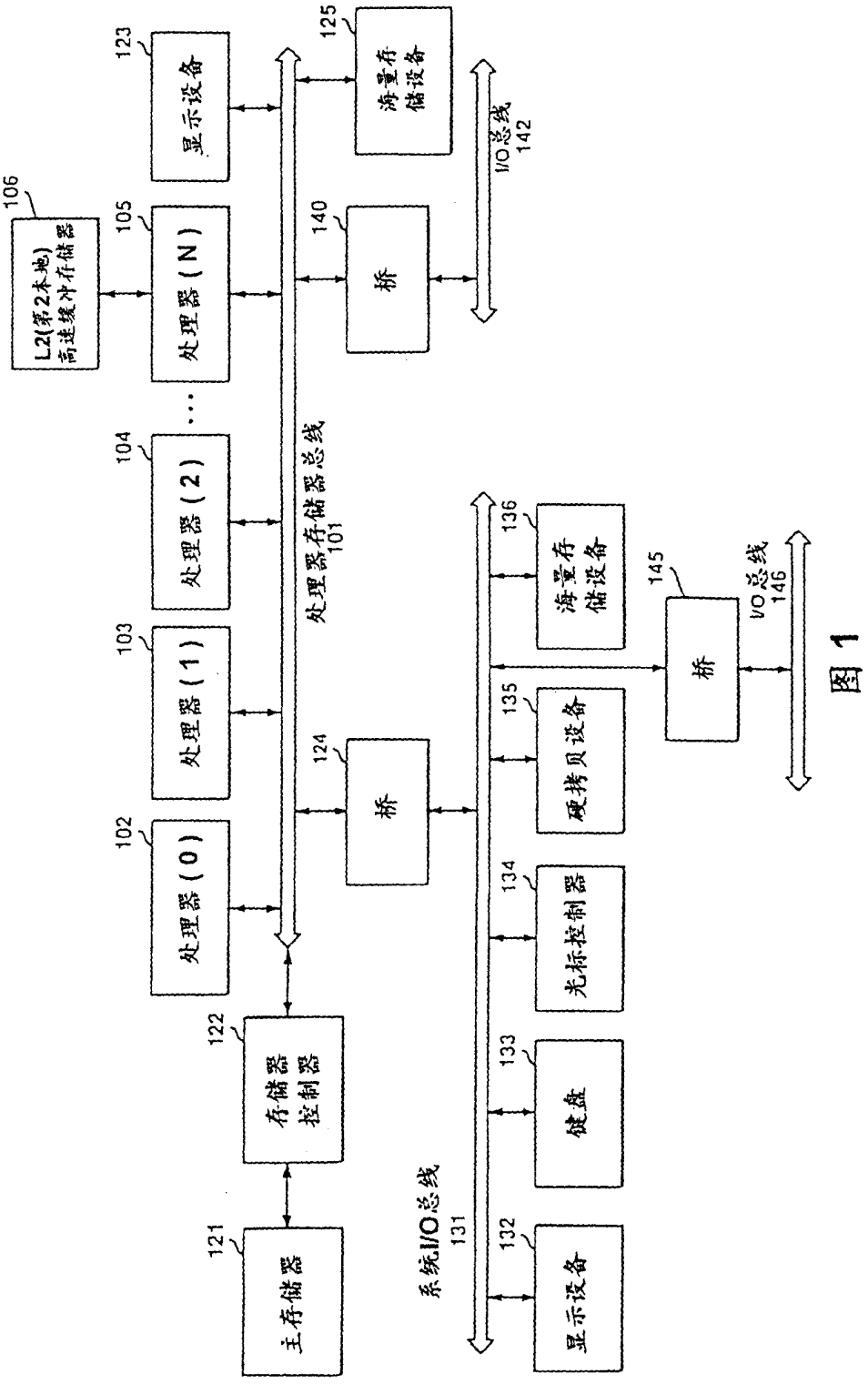

L2(第2本地)高速缓冲存储器
106
处理器(0)
102
处理器(1)
103
处理器(2)
104
处理器(N)
105
显示设备
123
海量存储设备
125
桥
140
I/O总线
142
处理器存储器总线
101
桥
124
存储器控制器
122
主存储器
121
系统I/O总线
131
显示设备
132
键盘
133
光标控制器
134
硬拷贝设备
135
海量存储设备
136
桥
145
I/O总线
146

图 1

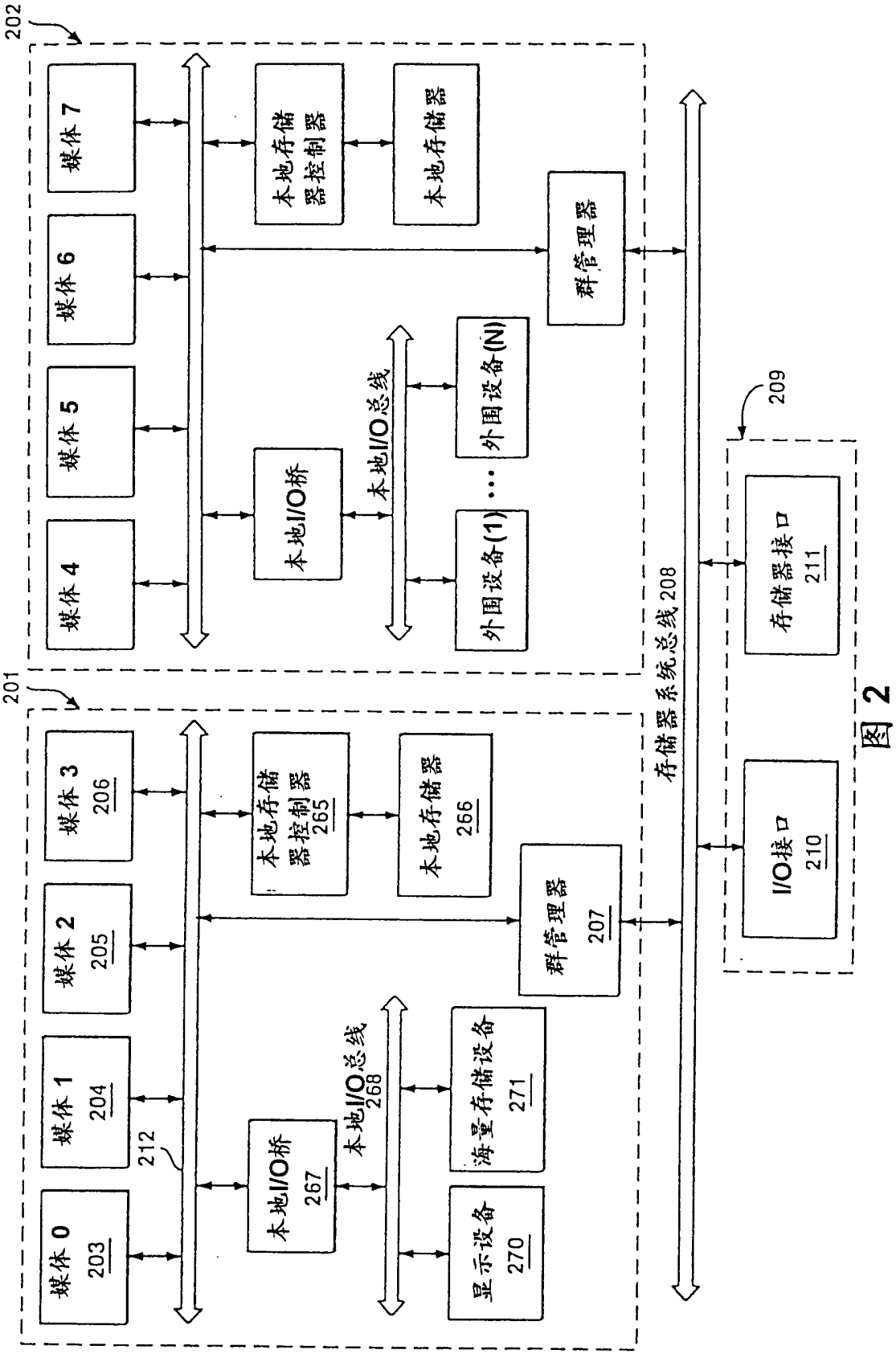

202
媒体 7
媒体 6
媒体 5
媒体 4
本地存储器控制器
本地存储器
本地I/O桥
本地I/O总线
外围设备(N)
外围设备(1)
群管理器
201
媒体 3
206
媒体 2
205
媒体 1
204
媒体 0
203
212
本地存储器控制器 265
本地存储器 266
群管理器 207
本地I/O桥 267
本地I/O总线 268
海量存储设备 271
显示设备 270
存储器系统总线 208
209
存储器接口 211
I/O接口 210

图 2

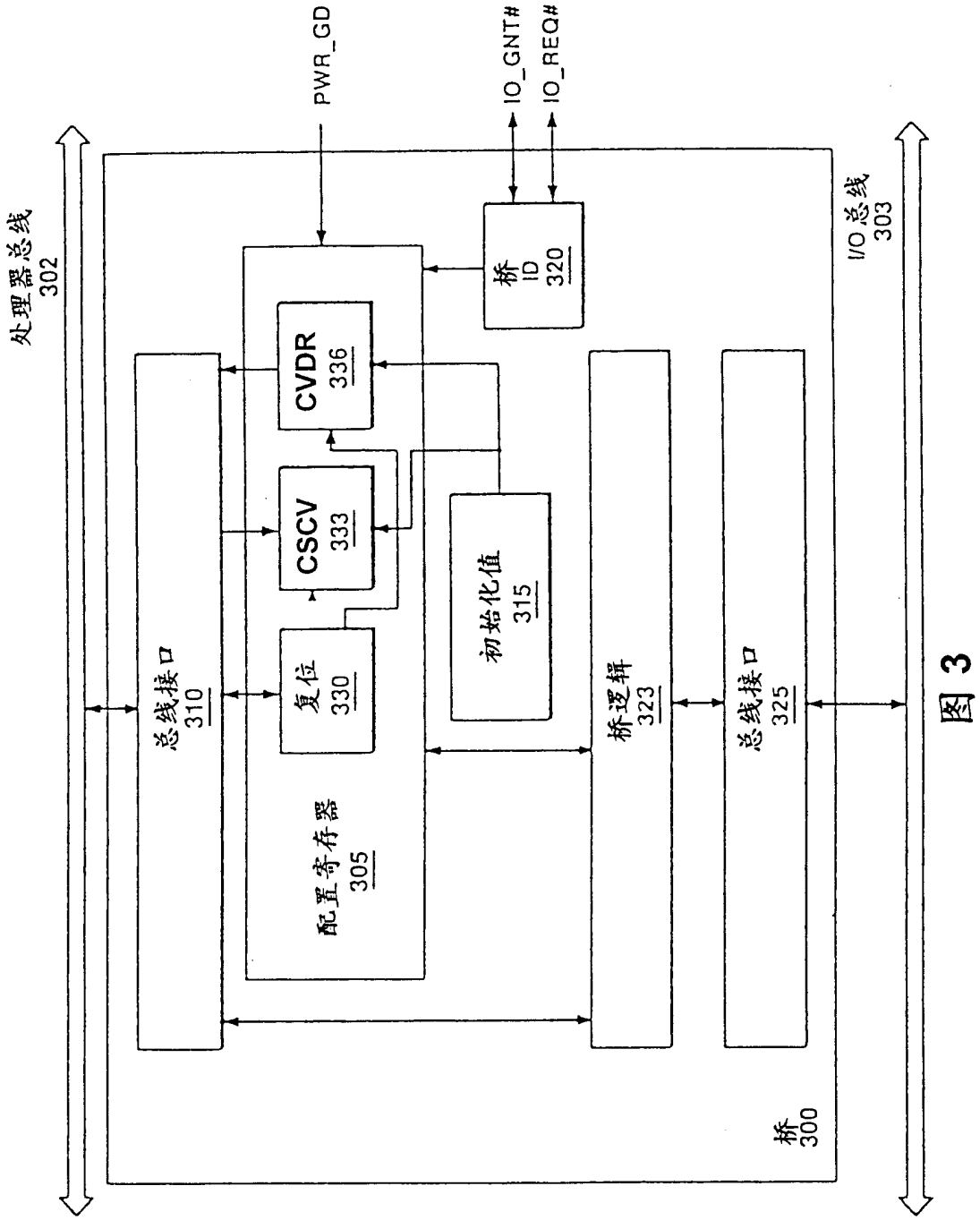

处理器总线
302
PWR_GD
IO_GNT#
IO_REQ#
I/O 总线
303
桥
ID
320
CVDR
336
CSCV
333
复位
330
初始化值
315
总线接口
310
配置寄存器
305
桥逻辑
323
总线接口
325
桥
300

图 3

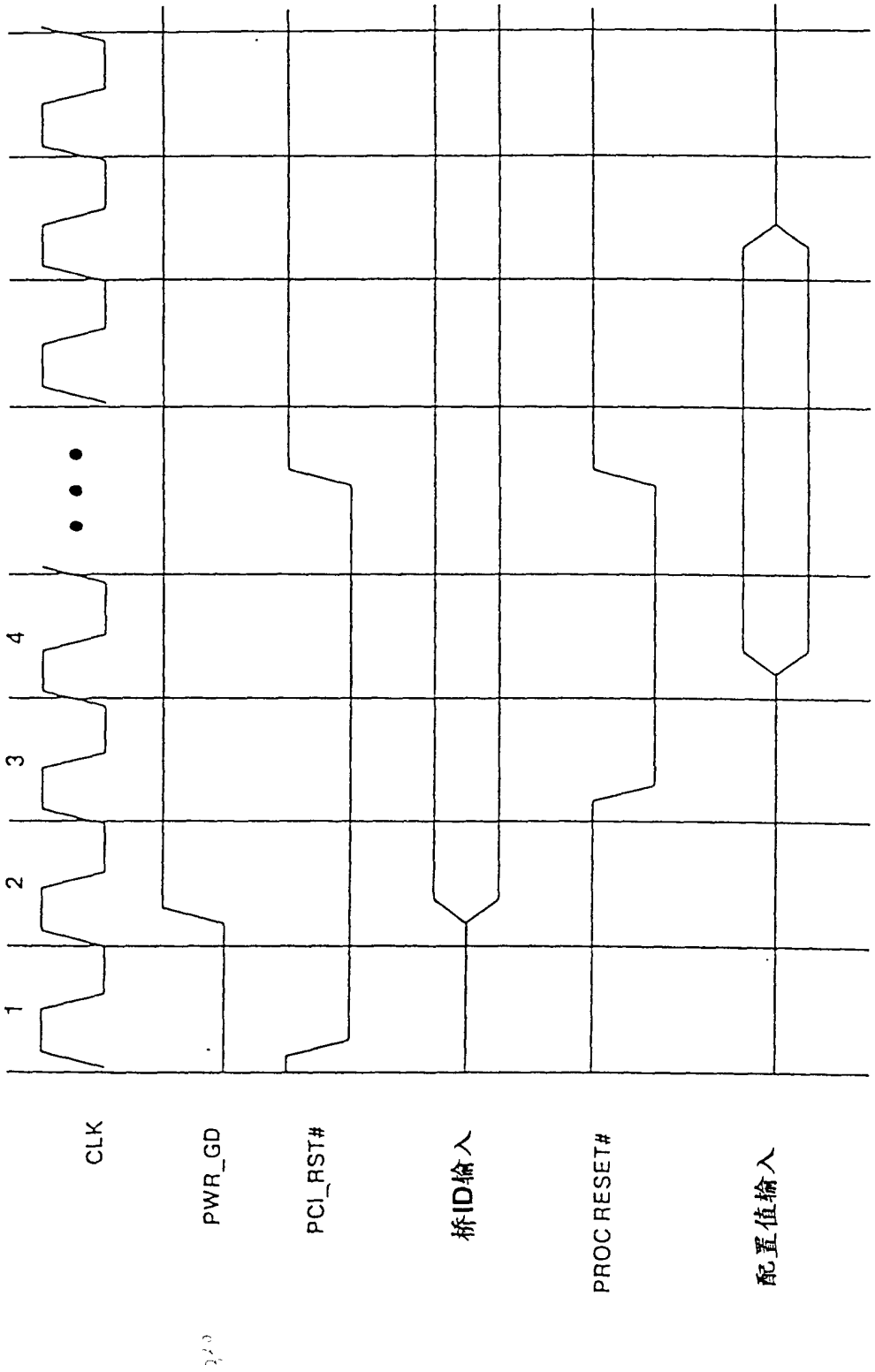


图 4

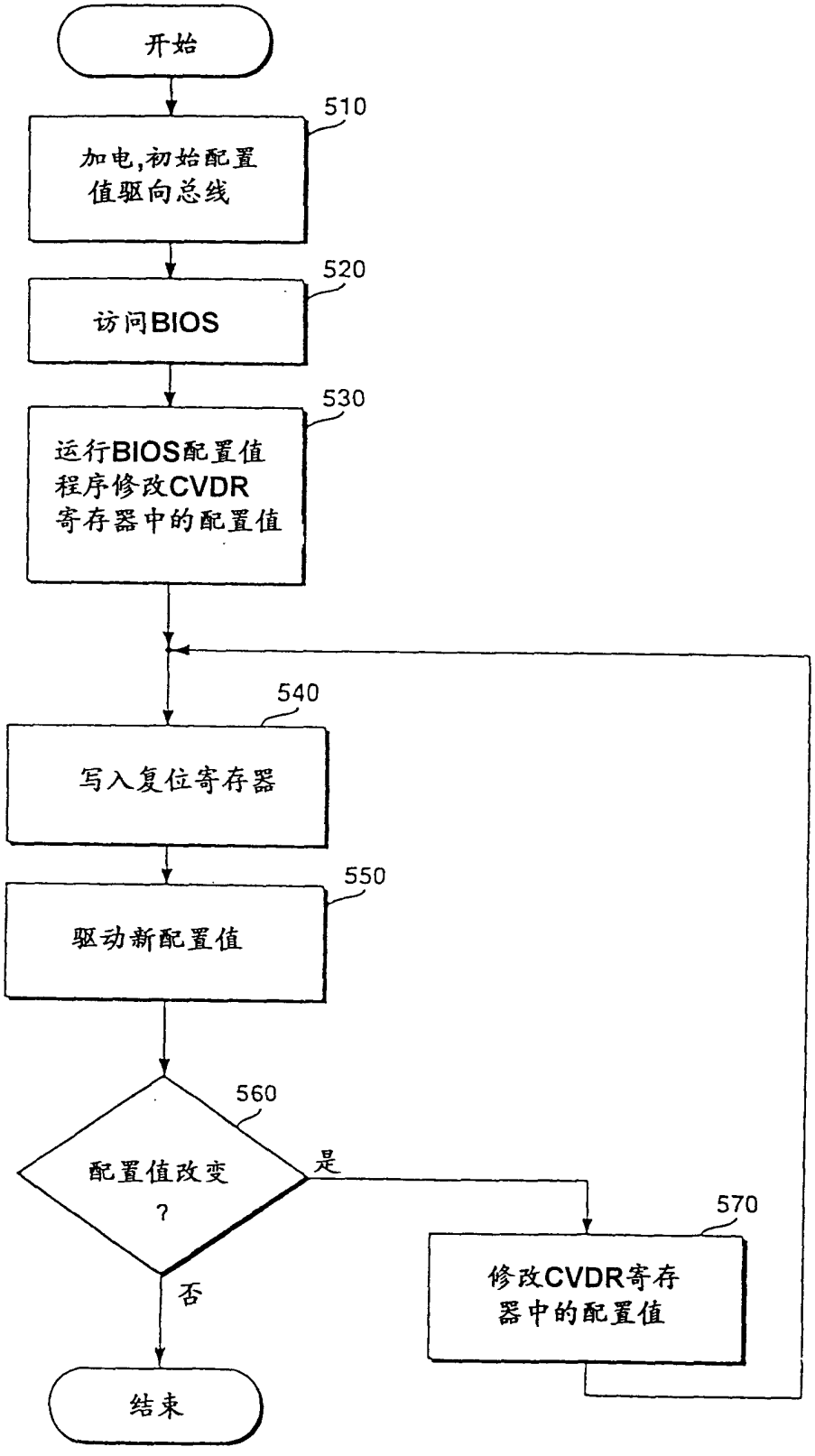

开始
510
加电,初始配置值驱向总线
520
访问BIOS
530
运行BIOS配置值程序修改CVDR寄存器中的配置值
540
写入复位寄存器
550
驱动新配置值
560
配置值改变?
是
否
570
修改CVDR寄存器中的配置值
结束

图 5