



## (12) 发明专利

(10) 授权公告号 CN 103207842 B

(45) 授权公告日 2016. 04. 20

(21) 申请号 201310110881. 8

US 6163834 A, 2000. 12. 19,

(22) 申请日 2006. 12. 18

CN 1531303 A, 2004. 09. 22,

(30) 优先权数据

11/323262 2005. 12. 30 US

Ram Huggahalli, Ravi Iyer, Scott

Tetrick. Direct Cache Access for High Bandwidth Network I/O. 《ISCA '05 Proceedings of the 32nd annual international symposium on Computer Architecture》. 2005, 第 33 卷 (第 2 期), 50 — 59.

(62) 分案原申请数据

200680050112. 1 2006. 12. 18

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

审查员 李艳军

(72) 发明人 R. S. 马杜卡鲁穆库马纳

S. 穆思拉萨纳卢尔 R. 哈加哈利

R. 伊利卡尔

(74) 专利代理机构 中国专利代理 (香港) 有限公司 72001

代理人 曲卫涛 卢江

(51) Int. Cl.

G06F 13/16(2006. 01)

(56) 对比文件

CN 1609826 A, 2005. 04. 27,

US 6725289 B1, 2004. 04. 20,

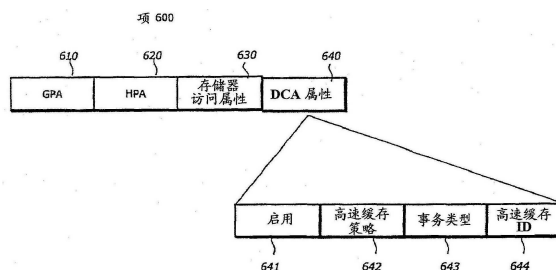
权利要求书2页 说明书6页 附图7页

(54) 发明名称

用于执行直接高速缓存访问事务的处理器和系统

(57) 摘要

公开用于基于存储器访问数据结构对直接高速缓存访问事务进行编码的装置、方法和系统的实施例。在一个实施例中, 装置包括存储器访问逻辑和事务逻辑。存储器访问逻辑用于基于存储器访问数据结构确定是否允许存储器访问。事务逻辑用于基于存储器访问数据结构为事务指定直接高速缓存访问属性。



1. 一种用于执行直接高速缓存访问事务的处理器,包括:  
存储器访问逻辑,所述存储器访问逻辑基于存储器访问数据结构来确定是否允许存储器访问;以及  
事务逻辑,所述事务逻辑基于所述存储器访问数据结构为事务指定直接高速缓存访问属性。
2. 如权利要求 1 所述的处理器,其中所述事务逻辑还基于所述直接高速缓存访问属性来对所述事务进行编码。
3. 如权利要求 1 所述的处理器,其中所述存储器访问逻辑还基于所述存储器访问数据结构转换存储器地址。
4. 如权利要求 1 所述的处理器,其中所述存储器访问逻辑还基于所述存储器访问数据结构将访客地址转换为主机地址。
5. 如权利要求 1 所述的处理器,还包括用于存储所述存储器访问数据结构的一部分的缓冲器。
6. 如权利要求 1 所述的处理器,其中所述直接高速缓存访问属性包括直接高速缓存访问启用指示符、高速缓存策略指示符、事务类型指示符、目标高速缓存识别指示符其中之一。
7. 一种用于执行直接高速缓存访问事务的系统,包括:  
从存储器访问数据结构获得直接高速缓存访问事务的属性的第一逻辑;以及  
为直接高速缓存事务指定所述属性的第二逻辑。
8. 如权利要求 7 所述的系统,还包括在所述存储器访问数据结构中存储所述属性的第三逻辑。
9. 如权利要求 7 所述的系统,还包括基于所述属性对处理器总线上的事务进行编码的第四逻辑。
10. 如权利要求 7 所述的系统,还包括基于所述存储器访问数据结构转换存储器地址的第五逻辑。
11. 如权利要求 7 所述的系统,其中所述属性包括直接高速缓存访问启用指示符、高速缓存策略指示符、事务类型指示符和目标高速缓存识别指示符之一。
12. 一种用于执行直接高速缓存访问事务的系统,包括:  
处理器,所述处理器包括基于存储器访问数据结构确定是否允许存储器访问的存储器访问逻辑和基于所述存储器访问数据结构为事务指定直接高速缓存访问属性的事务逻辑;  
以及  
耦合到所述处理器的系统存储器。
13. 如权利要求 12 所述的系统,其中所述事务逻辑还基于所述直接高速缓存访问属性来对所述事务进行编码。
14. 如权利要求 12 所述的系统,其中所述存储器访问逻辑还基于所述存储器访问数据结构转换存储器地址。
15. 如权利要求 12 所述的系统,其中所述存储器访问逻辑还基于所述存储器访问数据结构将访客地址转换为主机地址。
16. 如权利要求 12 所述的系统,还包括用于存储所述存储器访问数据结构的一部分的

缓冲器。

17. 如权利要求 12 所述的系统,其中所述直接高速缓存访问属性包括直接高速缓存访问启用指示符、高速缓存策略指示符、事务类型指示符、目标高速缓存识别指示符其中之一。

## 用于执行直接高速缓存访问事务的处理器和系统

### 技术领域

[0001] 本公开涉及信息处理领域,更具体地说,涉及信息处理系统中输入/输出设备与存储器之间的事务的领域。

### 背景技术

[0002] 直接高速缓存访问“DCA”是允许将来自输入/输出(“I/O”)设备的数据置于处理器的高速缓存中的信息处理系统协议。可以使用 DCA,通过在将数据置于系统存储器之前、替代将数据置于系统存储器、或与将数据置于系统存储器并行地将数据置于处理器的高速缓存中,或通过将数据置于系统存储器或中间高速缓存中并利用预取提示来触发将数据置于处理器的高速缓存中,来避免系统存储器访问等待时间和带宽限制。

[0003] DCA 协议涉及:使用指定给每个 I/O 事务的 DCA 属性,这些 DCA 属性可以确定是对特定事务启用还是禁用 DCA;将把数据置于哪个处理器高速缓存中;如何将数据置于处理器的高速缓存中(例如,置于多路高速缓存的哪一路中以及在什么样的行状态,如独占型或修正型);或 DCA 事务的其它细节。这些属性中的一些属性可以是实现或平台特有的,例如,在预取方法中,DCA 属性可以包括多行预取的长度属性或预取跨距(stride)。

[0004] 在外围总线事务中,通常将 DCA 属性从 I/O 设备传递给芯片集,该芯片集对处理器总线上的对应事务进行编码以将数据传递给处理器的高速缓存。例如,图 1 示出用于执行系统 100 中的 DCA 事务的已知方法,在系统 100 中,处理器 110 和 120 通过处理器总线 105 耦合到芯片集 130。芯片集 130 通过外围总线 135 耦合到 I/O 设备 140。处理器 110 包括核 112 和高速缓存 113、核 114 和高速缓存 115、核 116 和高速缓存 117、以及核 118 和高速缓存 119。处理器 120 包括核 122 和高速缓存 123、核 124 和高速缓存 125、核 126 和高速缓存 127、以及核 128 和高速缓存 129。

[0005] 系统 100 中的 DCA 事务包括外围总线 135 上的 I/O 事务 150,其中 I/O 事务 150 包括用于指定地址和/或数据的字段 151 和用于指定事务的 DCA 属性的字段 152。芯片集 130 接收 I/O 事务 150,将字段 151 和 152 的内容解码,并且如果需要,还对系统事务 160 进行编码。系统事务 160 包括用于指定地址和/或数据的字段 161 和用于指定事务的 DCA 属性的字段 162。在图 1 中,系统事务 160 在处理器总线 105 上执行,并将目标锁定在高速缓存 115 上。

[0006] 在不同的信息处理系统中使用不同的外围总线(如外围组件互连(“PCI”)、PCI-X、PCI-Express 和通用串行总线(“USB”))可能需要不同的方法来传递 I/O 事务中的 DCA 属性。例如,因为当定义 PCI-Express 时不存在 DCA,所以不存在独占地分配用于 DCA 使用的 PCI-Express 字段。因此,针对 PCI-Express 总线上的 DCA 事务的常规方法可以例如使用 PCI-Express 事务中的“标签”字段的有限的 5 位部分。

[0007] 在不同 I/O 总线上使用此类不同的方法而不是标准方法可能限制 DCA 的广泛采用。

## 附图说明

- [0008] 附图中举例而不是限制性地示出本发明。
- [0009] 图 1 示出执行 DCA 事务的已知方法。
- [0010] 图 2 示出根据本发明的实施例可以在其中执行直接高速缓存访问事务的系统。
- [0011] 图 3 示出对域的 I/O 设备指定的实例。
- [0012] 图 4 示出利用 DMA 重新映射来支持虚拟化的实例。
- [0013] 图 5 示出 I/O 地址转换的实例。
- [0014] 图 6 示出根据本发明的实施例的存储器访问数据结构中的项。
- [0015] 图 7 示出根据本发明的实施例的用于执行直接高速缓存访问事务的方法。

## 具体实施方式

[0016] 本发明可以在用于基于如下所述的存储器访问数据结构执行 DCA 事务的装置、方法或系统中实施。在以下描述中,阐述了诸如逻辑、电路和系统配置的众多具体细节,以便更充分地理解本发明。但是,本领域的技术人员将明白,在没有这些具体细节的情况下,也可以实现本发明。此外,没有详细示出一些熟知的结构、电路、技术等,以免不必要地使本发明晦涩难懂。

[0017] 本发明的实施例的元件可以在硬件、软件、固件或硬件、软件或固件的任意组合中实现。术语硬件一般是指具有物理结构的元件,如电子、电磁、光、电-光、机械、电-机部件等。术语软件一般是指逻辑结构、方法、过程、程序、例行程序、进程、算法、公式、表达式等。术语固件一般是指在硬件结构(如闪存存储器或只读存储器)中实现或实施的逻辑结构、方法、过程、程序、例行程序、进程、算法、公式或表达式。固件的实例有微代码、可写控制存储结构和微程序控制结构。

[0018] 图 2 是根据本发明的实施例的可以在其中执行直接高速缓存访问事务的系统 200 的图示。系统 200 包括处理器 210 和 220、处理器总线 205、存储器控制集线器(“MCH”)230、系统存储器 240、输入/输出控制集线器(“ICH”)250、外围总线 255、大容量存储设备/接口 270、以及 I/O 设备 280<sub>1</sub>-280<sub>k</sub>和 285。系统 200 可以包括比上述元件更多或更少的元件。

[0019] 处理器 210 和 220 可以是具有一个或多个执行核的任何组件,其中每个执行核可以基于各种不同类型的处理器中的任何一种处理器,例如来自英特尔公司的 **Pentium®** 处理器系列、**Itanium®** 处理器系列或其它处理器系列中的处理器,或来自其它公司的任何其它通用或其它处理器。处理器 210 包括核 212、214、216 和 218 以及高速缓存 213、215、217 和 219。处理器 220 包括核 222、224、226 和 228 以及高速缓存 223、225、227 和 229。

[0020] 处理器总线 205 可以是允许处理器 210 和 220 相互通信或与其它处理器或其它设备(如 MCH230)通信的任何并行、连续、流水线、异步、同步或其它总线或点对点连接。

[0021] MCH230 提供对分别诸如系统存储器 240 和 ICH250 的存储器和 I/O 设备的控制和配置。MCH230 包括如下所述的用于与处理器总线 205 对接并对它上面的事务进行编码的总线接口单元 231 和 DMA 重新映射逻辑 235。MCH230 可以集成到芯片集上,该芯片集还集成其它组件(如 ICH250)和/或多种功能性(如系统逻辑、总线桥接、存储器控制和 I/O 设备控制)。

[0022] 系统存储器 240 可以是用于存储诸如程序代码和数据的信息的任何类型的存储

器。系统存储器 240 通常用动态随机存取存储器 (“DRAM”) 或静态随机存取存储器 (“SRAM”) 来实现。系统存储器 240 可用于存储代码、数据或任何其它信息,它包括操作系统 (“OS”) 或虚拟机监视器 (“VMM”) 242 的整体或部分、I/O 驱动程序 245、以及用于实现本发明的实施实施例的任何代码或数据,如存储器访问表 248,如下所述。

[0023] 大容量存储设备 / 接口 270 可以对接到任何类型的磁、光、或其它机器可访问介质,如光盘 272、数字多用光盘 273、软盘 274 和硬盘驱动器 276。代码、数据和包括本发明的实施实施例的任何其它信息可以存储在任何这样的介质上并通过大容量存储设备 / 接口 270 访问。

[0024] ICH250 可以包括多种外围功能,例如直接存储器访问 (“DMA”) 控制器、中断控制器、功率管理逻辑、计时器、I/O 设备接口、以及诸如 USB 或低引脚数 (“LPC”) 接口的至外围总线 255 和 / 或任何其它 总线或连接的接口。ICH250 可以与诸如 MCH230 的任何其它组件一起集成到芯片集中。

[0025] ICH250 直接对接到 I/O 设备 285 并通过外围总线 255 对接到 I/O 设备 280<sub>1</sub>-280<sub>k</sub>。外围总线 255 可以是支持 I/O 事务的任何类型的总线,如 PCI、PCI Express 或工业标准体系结构 (“ISA”) 总线。I/O 设备 280<sub>1</sub>-280<sub>k</sub> 和 285 可以是用于执行包括 DMA 请求在内的 I/O 功能的任何 I/O 设备,如输入设备控制器 (如键盘、鼠标、跟踪球、定位设备)、介质卡 (如音频、视频、图形卡)、网络卡和任何其它外围设备。

[0026] 返回到 MCH230, MCH230 包括 DMA 重新映射逻辑 235, 该重新映射逻辑 235 在 I/O 事务中将 I/O 设备 (如 I/O 设备 280<sub>1</sub>-280<sub>k</sub> 和 285 之一) 映射到系统存储器 240 中的域 (如下所述)。I/O 事务通常是 DMA 请求 (即, 来自 I/O 设备的直接访问系统存储器 240 的请求), 但也可以是任何其它类型的事务。DMA 重新映射逻辑 235 提供硬件支持以促进或增强 I/O 设备指定和 / 或管理。或者, DMA 重新映射逻辑 235 可以包含在除 MCH230 以外的任何其它组件 (如 ICH250) 中。它也可以在处理器 210 或 220、或任何其它处理器、协处理器、芯片集或其它设备中部分或整体地实现。

[0027] 域抽象地定义为系统中的隔离环境, 为其分配系统存储器 240 的子集。被允许直接访问分配给一个域的存储器的 I/O 设备称为指定给那个域的设备。域的隔离是通过阻止没有指定给它的设备访问分配给它的存储器来实现的。多个隔离域通过确保将所有 I/O 设备指定给某个域 (可能是默认域) 并限制所指定的每个设备只可访问分配给它的域的存储器而得以支持。

[0028] 每个域具有不同于存储器的系统视图的存储器视图或物理地址空间。用于根据存储器的系统视图访问存储器的地址称为主机物理地址 (“HPA”)。供一个域的资源用来访问它的物理地址空间的地址称为访客物理地址 (“GPA”)。如果一个域的 GPA 空间与 HPA 空间相同或是 HPA 空间的子集, 则认为该域不可重新定位。如果一个域的 GPA 空间经转换可以访问分配给它的 HPA 空间, 则认为该域可以重新定位。GPA 和 HPA 空间的逻辑分离提供加强存储器保护的基础。它需要地址保护, 并且可能需要可以证实由指定给域的设备生成的 GPA 并可能将它们转换为有效 HPA 的转换机制。DMA 重新映射逻辑 235 提供对称为 DMA 重新映射的这种功能性的硬件支持。

[0029] 图 3 是示出对域的 I/O 设备指定的实例的图。I/O 设备指定是指将 I/O 设备映射到系统存储器 240 中的域。映射由 DMA 重新映射逻辑 235 来支持。在图 3 中, 将设备 310

映射到系统存储器 240 中的域 340。域 340 可以具有用于设备 310 的一个或多个驱动程序 342。

[0030] DMA 重新映射逻辑 235 包括寄存器组 320、DMA 重新映射结构 322 和逻辑电路 324。寄存器组 320 包括多个寄存器,这些寄存器提供对供 DMA 重新映射结构 322、逻辑电路 324 和 I/O 设备的程序或驱动程序使用的控制或状态信息的存储。DMA 重新映射结构 322 提供在 GPA 到 HPA 的重新映射或地址转换时所用的基本结构、存储或表。逻辑电路 324 包括用于执行重新映射或地址转换操作的电路。

[0031] 可以采用多种方式来使用利用 DMA 重新映射逻辑 235 的 I/O 设备指定和 / 或管理,以便例如为 OS 提供隔离域。例如,OS 可以定义一个域来包含它的关键代码和数据结构,并限制所有 I/O 设备对该域的访问。这种域隔离技术允许 OS 限制通过 I/O 设备和它们的驱动程序对它的代码和数据进行错误、无意或其它讹用。

[0032] DMA 重新映射逻辑 235 还可用于支持虚拟化环境。一般来说,虚拟化环境是通过使用对系统和它的资源具有完全、直接控制的 VMM 软件创建系统 (“虚拟机”或“VM”) 的抽象来实现的,其中其它软件 (“访客”) 可以像是在对 VM 和它的资源的完全、直接控制下那样在这些系统上运行。VMM 通过在 VM 之间分配和共享系统资源并将 VM 相互隔离以保护每个 VM 免受在其它 VM 上运行的访客的影响来维护虚拟化环境。

[0033] DMA 重新映射逻辑 235 可用于通过将 VMM 映射到域、将每个 VM 映射到域并将各个 I/O 设备指定给对应于各个 VM 的各个域来支持虚拟化。因此,可以指定 I/O 设备的驱动程序只可在指定给它的 VM 中运行,并且可以允许该驱动程序与 I/O 设备直接交互,而不必向 VMM 传递控制。

[0034] 图 4 示出利用 DMA 重新映射来支持虚拟化的实例。在图 4 中,在 VM440 上运行的访客包括设备驱动程序 450、访客 OS446 以及应用程序 442 和 444。在 VM460 上运行的访客包括设备驱动程序 470、访客 OS466 以及应用程序 462 和 464。设备驱动程序 450 和 470 分别是驱动、控制、对接、或支持 I/O 设备 410 和 412 的驱动程序。I/O 设备 410 指定给 VM440,而 I/O 设备 412 指定给 VM460。

[0035] 一般来说,访客 (guest) 通过例如在尝试 I/O 操作时陷入到 (trap to) VMM420 来通过 VMM420 访问系统资源。但是,由于设备 410 指定给 VM440 并且它的驱动程序 450 在 VM440 上运行,所以访客 OS446 和应用程序 442 与 444 可以不必陷入到 VMM420 便可访问设备 401。类似地,设备 410 可以不必陷入到 VMM420 便可通过例如 DMA 请求来访问分配给它的域的存储器。DMA 重新映射逻辑 235 通过将 I/O 设备所用的 GPA 转换为对应的 HPA 并拒绝 I/O 设备访问在分配给它的域之外的存储单元来支持这种能力。

[0036] 图 5 示出 I/O 地址转换的实例。在图 5 中,I/O 设备 510 指定给域 A,而 I/O 设备 512 指定给域 B。每个 I/O 设备可以执行对地址 DMA\_ADR 的 I/O 请求或 DMA 请求。DMA 重新映射逻辑 235 将来自 I/O 设备 510 的对 DMA\_ADR 的请求映射到物理存储器 520 的段 522,其中物理存储器的段 522 和 524 分配给域 A。类似地,DMA 重新映射逻辑 235 将来自 I/O 设备 512 的对 DMA\_ADR 的请求映射到物理存储器 520 的段 528,其中物理存储器的段 526 和 528 分配给域 A。段 522、524、526 和 528 可以根据系统的存储器管理模型的页的大小、或是物理存储器的一部分的任何其它大小。

[0037] 系统存储器的访客视图 540 是从访客 I/O 设备的角度看的逻辑视图。访客 I/O

设备 512 将段 542 看作是系统的地址空间,而访客 I/O 设备 510 将段 544 看作是系统的地址空间。负责创建和管理域的 VMM 或其它软件为每个域分配物理存储器 520,并在 DMA 重新映射逻辑 235 中建立 GPA-HPA 转换功能。

[0038] 通过 DMA 重新映射逻辑 235 实现的地址转换功能取决于系统的物理存储器管理模型。例如,在将主机物理存储器作为邻接区域分配给域时,转换功能可以是简单的偏移量相加。或者,DMA 重新映射逻辑 235 可能参考数据结构用于执行地址转换,例如在以页粒度管理主机物理存储器时。其它实施例可以使用数据结构来定义允许各个设备访问的地址范围。其它实施例也可以使用这些或其它技术的任意组合。

[0039] DMA 重新映射逻辑 235 所参考的数据结构(“存储器访问数据结构”)可以是任何形式的数据结构,如单级或多级表。数据结构或数据结构的任何部分可以存储在 DMA 重新映射逻辑 235 可访问的任何存储空间中,例如存储在 DMA 重新映射结构 322 中或存储在系统存储器 240 中的存储器访问表 248 中。数据结构中的所有或部分信息可以在一个或多个存储空间中进行拷贝或复制,例如,来自存储器访问表 248 的项或 DMA 重新映射逻辑 235 的转换结果可以存储在 DMA 重新映射结构 322 中的转换后备或其它缓冲器中或存储在其它位置。

[0040] 图 6 示出根据本发明的实施例的存储器访问数据结构中的项 600。项 600 包括 4 个字段,其中每个字段可以包括任何数量的位单元。GPA 字段 610 用于存储可供 I/O 设备用来访问系统存储器的 GPA 或 GPA 的一部分。根据如上所述的任何存储器地址转换功能,HPA 字段 620 用于存储对应于存储在 GPA 字段 610 中的 GPA 的 HPA 或 HPA 的一部分。在一个实施例中,存储在 GPA 字段 610 和 HPA 字段 620 中的每个地址指定系统存储器的页的基本地址,以使得对系统存储器中的单个页的所有访问只需存储器访问数据结构中的一个项。在此类实施例中或在其它实施例中,存储器访问数据结构中的单个项可以表示地址范围而不是单个地址。

[0041] 存储器访问属性字段 630 用于存储对应于字段 610 和 620 中的项的存储器访问属性。例如,存储器访问字段 630 可用于存储指示是否允许对对应地址或地址范围的 DMA(或其它 I/O)事务的值、指示是否必须在对对应地址或地址范围的 DMA(或其它 I/O)事务上窥探处理器高速缓存的值、以及与对对应地址或地址范围的 DMA(或其它 I/O)事务有关的任何其它值或指示符。

[0042] DCA 属性字段 640 用于存储对应于字段 610 和 620 中的项的 DCA 属性。例如,DCA 属性字段 640 可以包括:启用/禁用字段 641,用于存储指示对对应地址或地址范围是启用还是禁用 DCA 的值;高速缓存策略字段 642,用于存储指示供对对应地址或地址范围的 DCA 事务使用的高速缓存策略(例如,独占型或修正型)的值;系统事务类型字段 643,用于存储指示供对对应地址或地址范围的 DCA 事务使用的系统事务类型的值;目标字段 644,用于存储指示待用作对对应地址或地址范围的 DCA 事务的目标的处理器或高速缓存的身份的值;以及任何其它字段,用于存储与对对应地址或地址范围的 DCA 事务有关的值、指示符或属性(例如,在预取方法中,指多行预取的长度属性或预取跨距)。

[0043] 图 7 示出根据本发明的实施例用于执行系统 200 中的 DCA 事务的方法 700。在方框 710,OS(或 VMM)242 配置 DMA 重新映射逻辑 235,并根据 DMA 重新映射逻辑 235 的特定实现和所需的地址保护和/或转换功能来建立存储器访问数据结构 248。



[0044] 在方框 720, I/O 驱动程序 245 向 OS (或 VMM) 242 做出对存储器访问数据结构 248 中的项 (如项 500) 启用 DCA 的请求。或者, 该项可以是 DMA 重新映射逻辑 235 或系统 200 中的任何其它存储空间中的存储器访问数据结构内的项。在做出请求时, I/O 驱动程序 245 提供待存储在项 600 中的所需的 DCA 属性。负责提供 I/O 驱动程序 245 用来做出 DCA 请求并提供 DCA 属性的软件接口的 OS (或 VMM) 242 处理该请求。因此, 项 600 内的整个 DCA 属性集可以包括由 I/O 驱动程序 245 提供的属性和由 OS (或 VMM) 242 提供的属性。作为方框 710 中的存储器访问数据结构 248 的建立的一部分, 作为系统 200 的任何随后配置或在系统 200 中执行的处理的一部分, 或者通过 OS (或 VMM) 242 进行的作为处理在方框 720 中的来自 I/O 驱动程序 245 的请求的一部分, 可以将对应的 GPA 和 HPA 值存储在项 600 中。

[0045] 在方框 730, 系统 200 中的 I/O 设备 (如 I/O 设备 180) 向由项 600 中的 GPA 指定的地址或地址范围发出外围总线 155 上的 I/O 事务。与参照图 1 描述的已知方法不同, I/O 设备不需要知道 DCA, 并且 I/O 事务可以是标准 I/O 事务, 如 DMA 存储器写事务, 从而不包含任何 DCA 属性。

[0046] 在方框 740, MCH230 接收来自方框 730 的 I/O 事务, 并且如果需要, 将该 I/O 事务解码。在方框 750, MCH230 参考项 600 以确定对项 500 中的 GPA 启用 DCA, 从项 600 获得 DCA 属性, 并且如果需要, 将 GPA 转换为 HPA。

[0047] 在方框 760, 总线接口单元 231 基于存储在项 600 中的 DCA 属性编码并发出处理器总线 105 上的对应于方框 730 中的 I/O 事务的事务, 以便将目标锁定为系统 200 中的高速缓存 (例如, 高速缓存 213、215、217、219、223、225、227 或 229 中的任何一个高速缓存)。

[0048] 根据本发明的实施例设计的 DMA 重新映射逻辑 235 或任何其它元件可以在从创建到模拟到制作的各个阶段进行设计。表示设计的数据可以用多种方式来表示这个设计。首先, 可以利用硬件描述语言或其它功能描述语言来表示硬件, 这在模拟中可用。作为补充或替代, 可以在设计过程的某些阶段产生具有逻辑和 / 或晶体管门的电路级模型。此外, 在某个阶段, 大多数设计达到能够用表示各种设备的物理布置的数据对其进行建模的程度。在使用常规半导体制作技术的情况下, 表示设备布置模型的数据可以是指定在用来制造集成电路的掩模的不同掩模层上各种特征存在与否的数据。

[0049] 在设计的任意表示中, 数据可以存储在任何形式的机器可读介质中。经调制或以其它方式生成用来传送此类信息的光波或电波、存储器、或诸如盘的磁或光存储介质都可以作为机器可读介质。所有这些介质都可“携带”或“指示”该设计、或本发明的实施例中所用的其它信息, 如错误恢复例行程序中的指令。当指示或携带信息的电载波得以传送时, 在执行电信号的复制、缓冲或重新传送的程度, 进行了新的复制。因此, 通信提供者或网络提供者的动作可以构成物品 (例如, 载波) 的复制, 从而实施本发明的技术。

[0050] 因此, 公开了用于基于存储器访问数据结构执行 DCA 事务的本发明的实施例。虽然已经描述并在附图中示出了某些实施例, 但是要了解, 这些实施例只是说明而不是限制宽广的本发明, 并且本发明并不局限于所示出和描述的具体结构和配置, 本领域的技术人员在研究本公开之后可以联想到各种其它修改。在发展迅速并且难以预见到进一步改进的诸如此类的技术领域, 在不背离本公开的原理或随附权利要求的范围的前提下, 通过促进技术进步可以容易地对所公开的实施例进行配置和细节方面的修改。

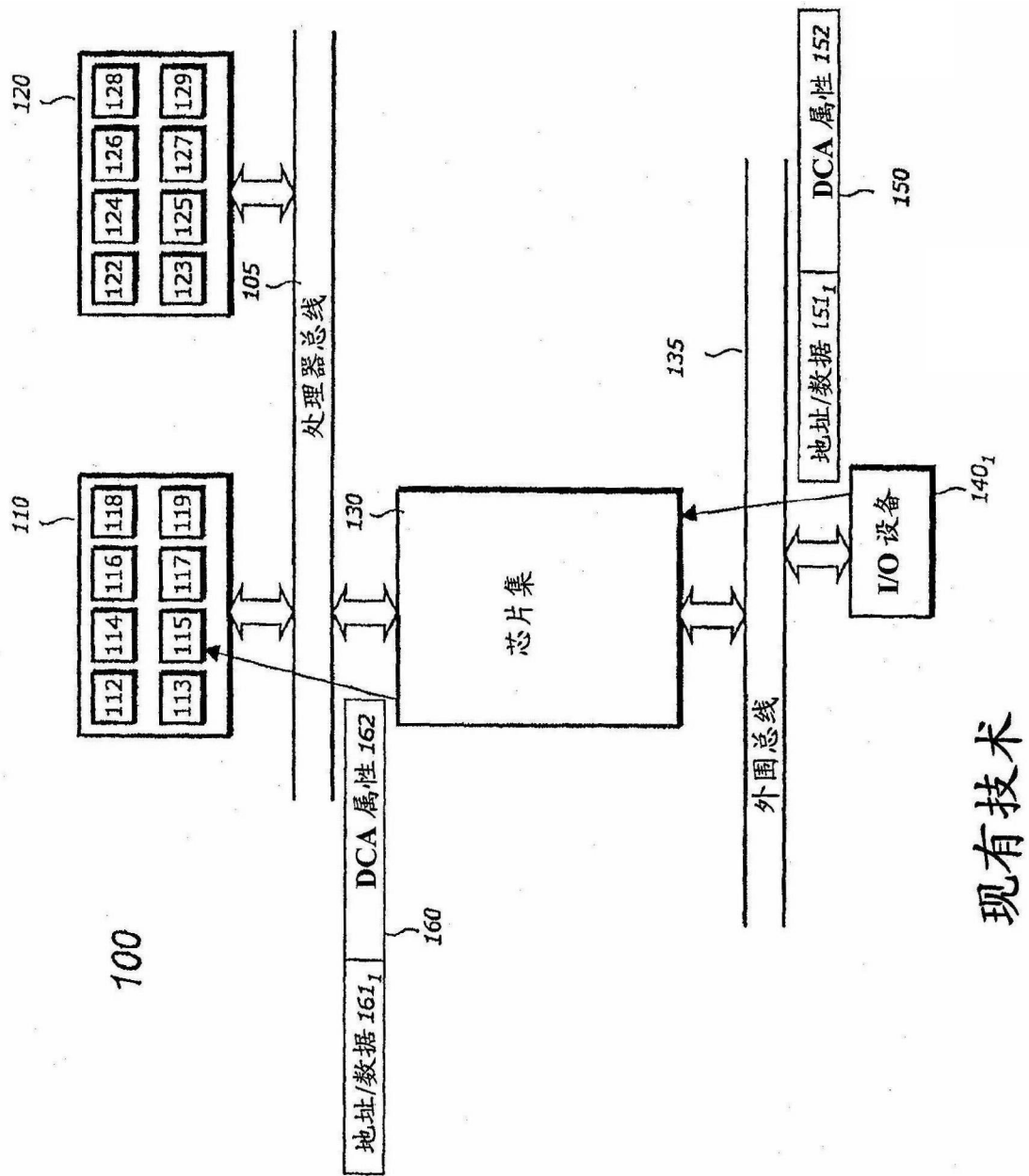


图 1

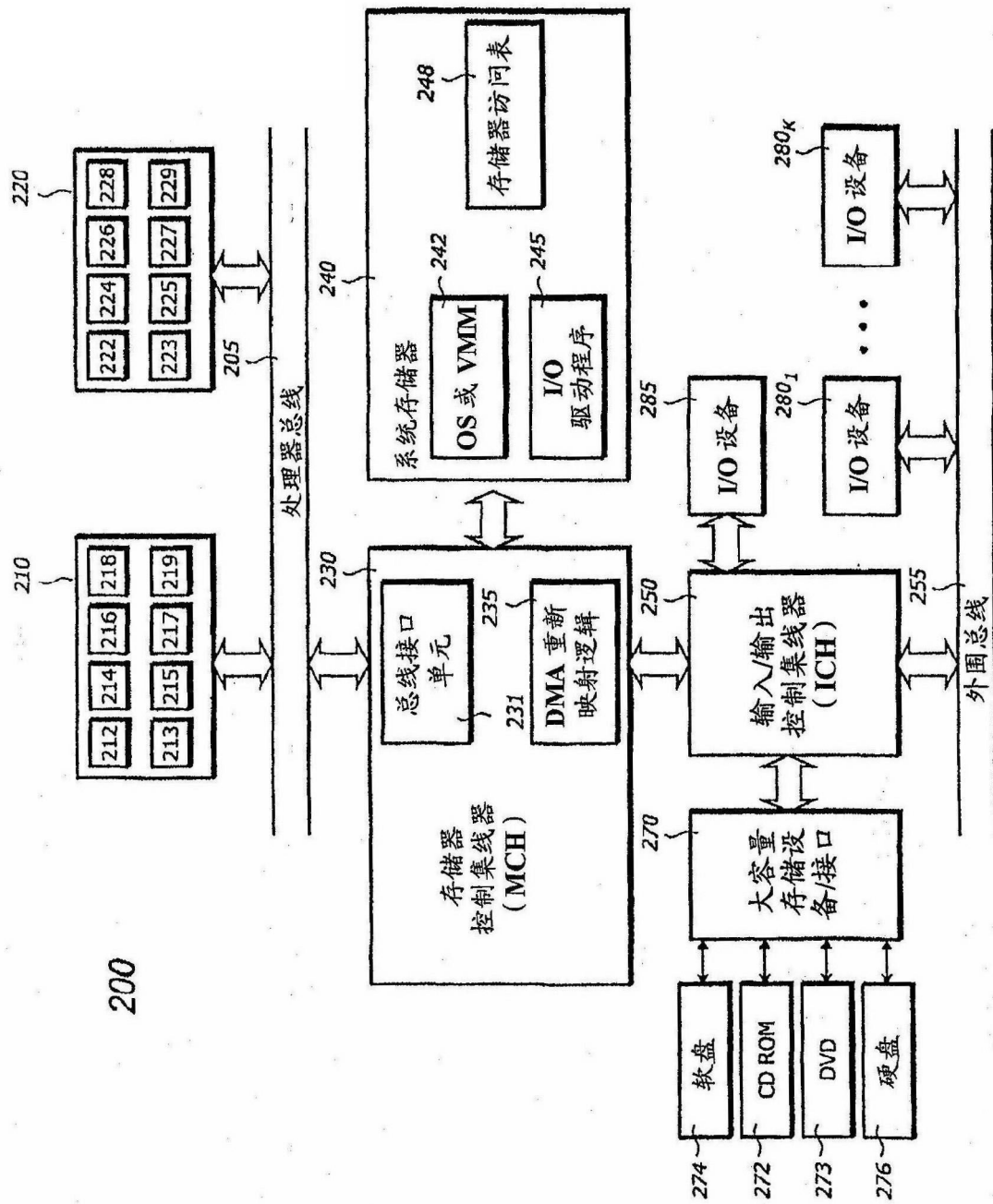


图 2

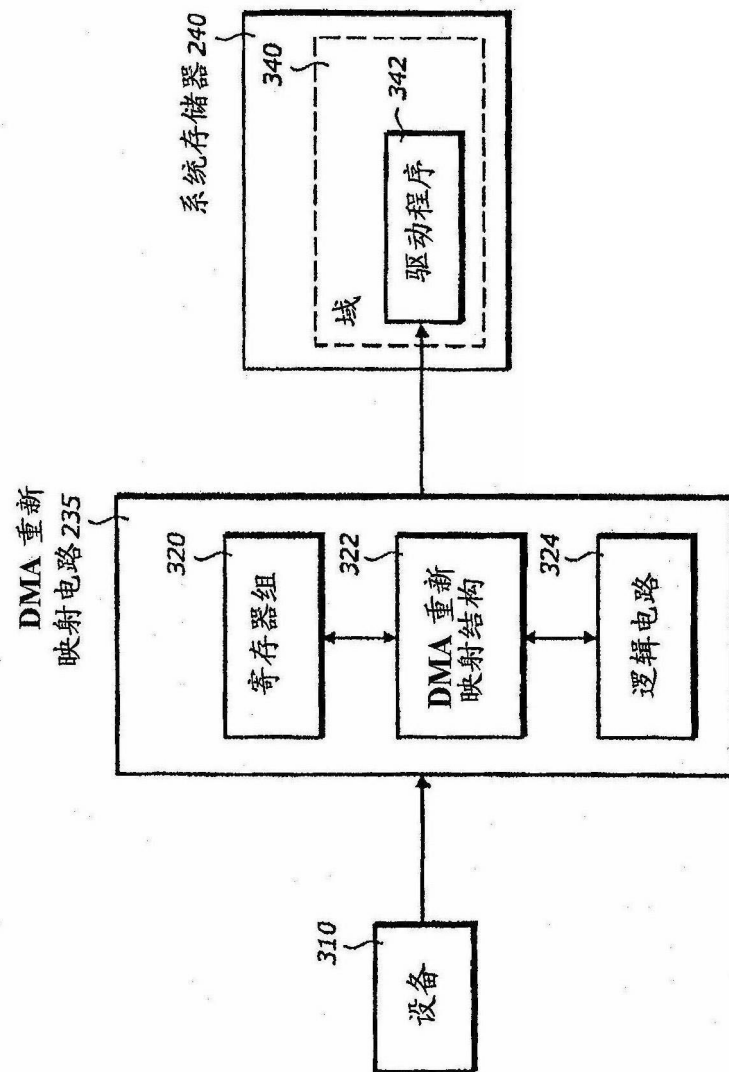


图 3

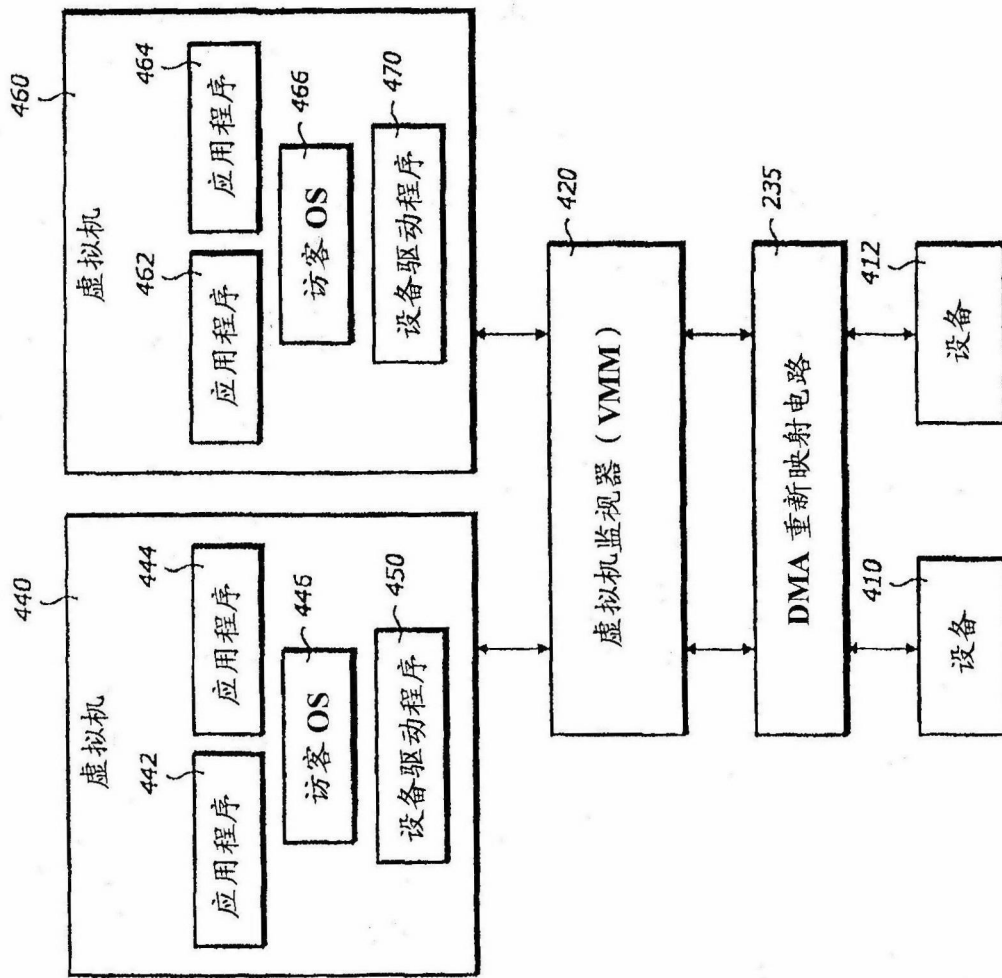


图 4

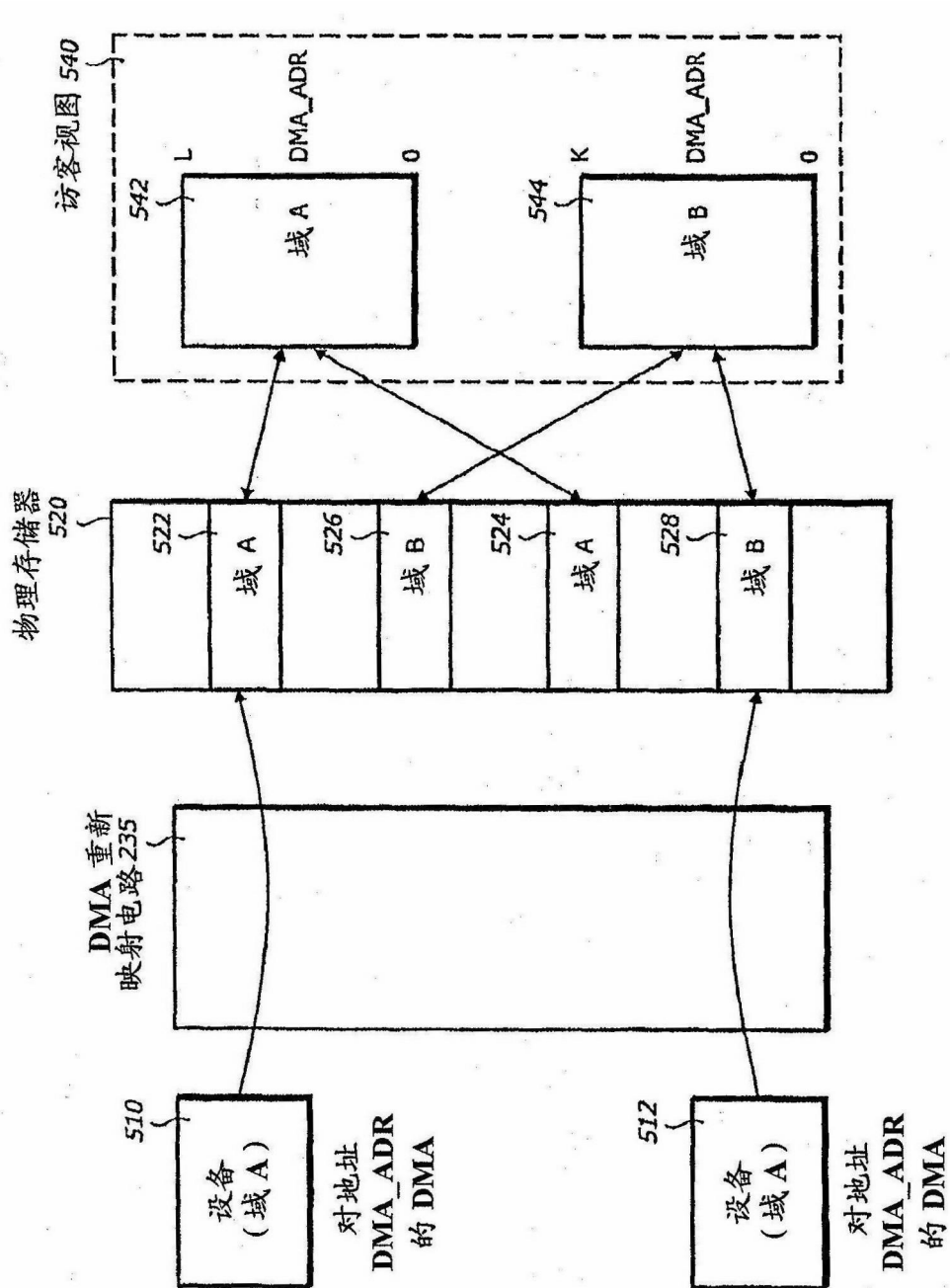


图 5

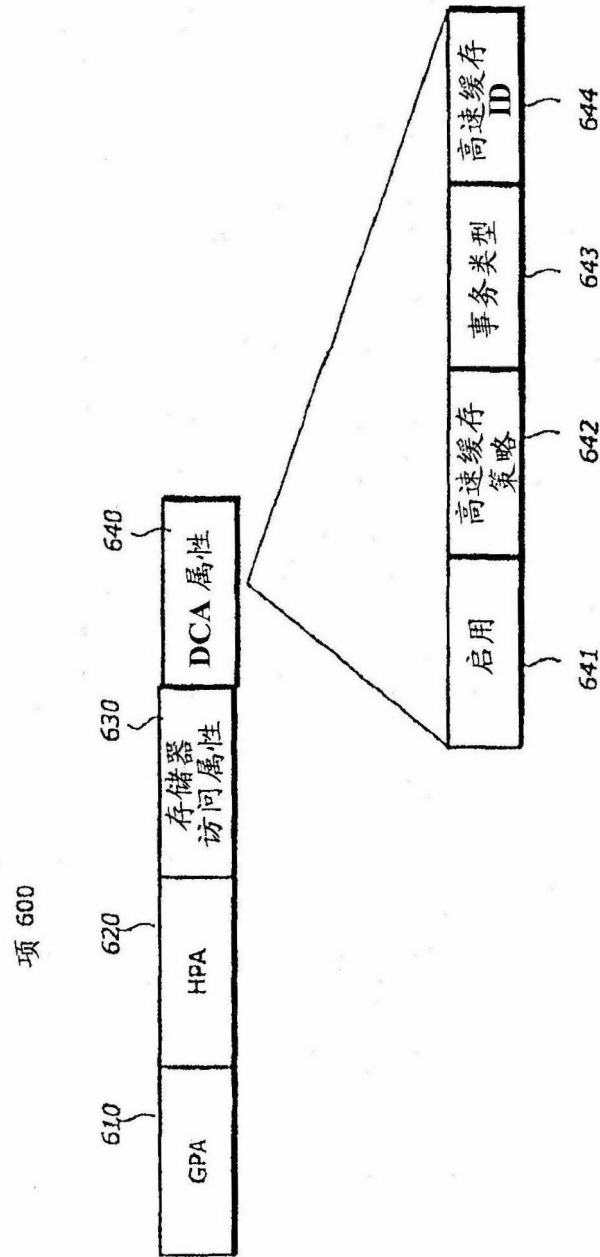


图 6

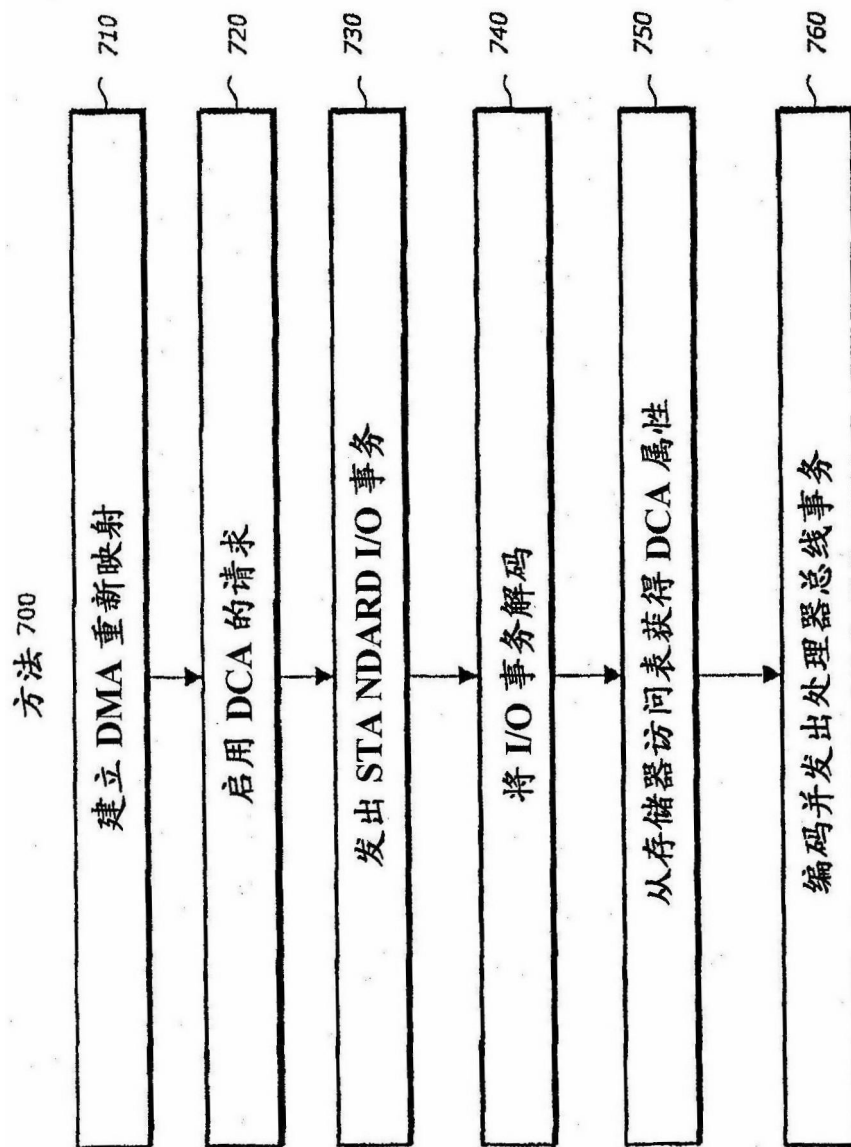


图 7