



[12] 发明专利说明书

专利号 ZL 200710110038.4

[45] 授权公告日 2009年8月26日

[11] 授权公告号 CN 100533416C

[22] 申请日 2007.6.12

[21] 申请号 200710110038.4

[30] 优先权

[32] 2006.8.15 [33] US [31] 11/464,613

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 古斯塔夫·E·希特曼三世

凯瑟琳·C·黄

丹尼尔·F·卡斯波

小弗兰克·W·布赖斯

约汉·R·弗拉纳甘

罗伯特·J·布里昂斯三世

马克·P·加蒂奈尔

约瑟夫·C·埃里奥特

[56] 参考文献

US5461721A 1995.10.24

US4368513A 1983.1.11

US5526484A 1996.6.11

审查员 刘邵频

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

代理人 申发振

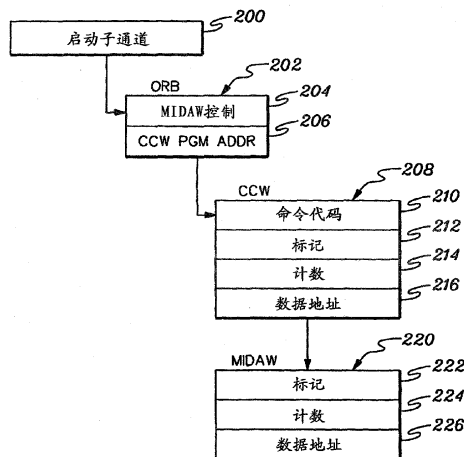
权利要求书2页 说明书14页 附图4页

[54] 发明名称

控制处理环境的数据传输的方法和系统

[57] 摘要

控制输入/输出设备和存储器之间的数据传输。数据传输在一个控制块的控制下开始，响应传输于一个控制块中指定的数据量，对于数据传输的控制从一个控制块传递到另一个控制块。控制的传递是独立于存储器边界进行的，从而在控制数据传输时具有灵活性。每一个控制块都包括控制和实现数据传输的字段。



1. 一种控制处理环境的数据传输的方法，所述方法包括：
在于一个控制块中指定的存储器地址开始数据传输；以及
响应于传输了在一个控制块中指定的数据量，将对于数据传输的控制独立于存储器边界的传递到另一个控制块。
2. 根据权利要求 1 所述的方法，其中，所述一个控制块包括表明要传输的数据量的计数。
3. 根据权利要求 1 所述的方法，进一步包括确定是否要传输另外的数据，其中，所述传递进一步响应于所述确定表明有另外的数据需要传输。
4. 根据权利要求 3 所述的方法，其中，所述确定包括将一个控制块的计数和与所述一个控制块关联的命令字的计数进行比较，以确定是否要传输另外的数据。
5. 根据权利要求 1 所述的方法，其中，另一个控制块包括将在传输中使用的存储器中的地址，所述地址指定不在指定存储器边界中并且与所述一个控制块指定的存储器的块不连续的存储器的块。
6. 根据权利要求 1 所述的方法，进一步包括，响应另一个控制块接收到控制，确定是否要跳过由所述另一个控制块指定的存储器的范围的数据传输，所述确定使用所述另一个控制块的指示器。
7. 根据权利要求 1 所述的方法，其中，所述一个控制块和所述另一个控制块包括在控制块的列表内，并进一步包括验证要在数据传输中使用的控制块的列表包括足够的控制块以传输指定的数据量。
8. 根据权利要求 7 所述的方法，其中，所述指定的数据量包括在与控制块的列表关联的命令字中，其中，所述验证使用控制块的列表的至少一个控制块的指示器。
9. 根据权利要求 1 所述的方法，其中，所述一个控制块和所述另一个控制块包括在控制块的列表中，所述控制块的列表是由程序的命令字指定的，用于实现数据传输，所述数据传输在一个或多个输入/

输出设备和非连续的存储器之间进行。

10. 根据权利要求 1 所述的方法, 其中, 数据传输不被处理环境的通道和控制单元之间的握手开销中断地进行。

11. 一种控制处理环境的数据传输的系统, 所述系统包括:

具有指定将开始数据传输的位置的存储器地址的一个控制块; 以及

另一个控制块, 响应于传输了在一个控制块中指定的数据量, 所述另一个控制块独立于存储器边界的从所述一个控制块接收数据传输的控制。

12. 根据权利要求 11 所述的系统, 其中, 所述一个控制块用于和与所述一个控制块关联的命令字的计数进行比较, 以确定是否要传输另外的数据。

13. 根据权利要求 11 所述的系统, 其中, 所述另一个控制块包括要在传输中使用的存储器中的地址的信息, 所述地址指定不在指定存储器边界中并且与所述一个控制块指定的存储器的块不连续的存储器的块。

14. 根据权利要求 11 所述的系统, 其中所述另一个控制块包括指示器, 用于响应所述另一个控制块接收控制, 确定是否要跳过由所述另一个控制块指定的存储器的范围的数据传输。

15. 根据权利要求 11 所述的系统, 其中, 所述一个控制块和所述另一个控制块包括在控制块的列表内, 其中, 所述控制块的列表的至少一个控制块包括指示器, 用于验证要在数据传输中使用的控制块的列表是否包括足够的控制块以传输指定的数据量。

16. 一种控制处理环境的数据传输的设备, 包括:

用于在于一个控制块中指定的存储器地址开始数据传输的装置; 以及

用于响应于传输了传输在一个控制块中指定的数据量, 将对于数据传输的控制独立于存储器边界的传递到另一个控制块的装置。

控制处理环境的数据传输的方法和系统

技术领域

一般而言，本发明涉及输入/输出处理，具体来说，涉及控制处理环境的输入/输出设备和存储器之间的数据传输。

背景技术

使用输入/输出 (I/O) 操作来在存储器和输入/输出设备之间传输数据。具体来说，通过执行输入/输出操作，将数据从存储器中写入到一个或多个输入/输出设备中，并将数据从一个或多个输入/输出设备中读取到存储器中。

为便于某些环境中的输入/输出操作的处理，使用通道命令字。通道命令字指定要执行的命令，对于启动某些 I/O 操作的命令，它指定与操作关联的存储区，每当从该存储区的传输或向该存储区的传输完成时要采取的操作，及其他选项。

通道命令字包括对主存储器进行直接定址或指定连续的地址列表的地址以提供间接定址的地址。连续的地址列表，称为间接数据定址字 (IDAW)，被用来控制跨非连续 2KB 或 4KB 存储器块的数据传输。特别是，当到达程序选择的 2KB 或 4KB 边界时，控制被传递到列表中连续的 IDAW。这种传送一直持续到已经传送了由通道命令字指定的总的数据量为止。由于将控制从一个 IDAW 传送到下一个 IDAW 的触发器是程序指定的 2KB 或 4KB 的存储器边界，由 IDAW 指定的存储器位置在程序指定的 2KB 或 4KB 边界上终止。

发明内容

基于上述内容，在输入/输出处理中需要更灵活的控制。例如，需要一种允许将对于数据传输的控制传递到程序指定的 2KB 或

4KB 边界之外的边界处的功能。此外，还需要一种允许待传输的存储器的块在任何边界处开始和结束，和/或大于 4KB 的功能。还进一步需要不被通道和控制单元之间的握手开销中断地进行数据传输的功能。另外还需要验证传输的数据量的功能。还需要其他控制选项。

通过提供控制处理环境中的数据传输的方法，克服了现有技术的缺点，并提供了另外的优点。该方法包括，例如，在于一个控制块中指定的存储器地址处开始数据传输；响应独立于指定存储器边界的传输在一个控制块中指定的数据量，将对于数据传输的控制传递到另一个控制块。

在一个实施例中，一个控制块和另一个控制块是修改的间接数据定址字。

作为本发明的另一个方面，提供了通道程序。通道程序包括，例如，指定在传输数据时可使用的多个修改的间接数据定址字的列表的通道命令字，其中，修改的间接数据定址字包括存储器地址和计数字段，其中，数据传输在列表中的一个修改的间接数据定址字的存储器地址处开始，响应独立于存储器边界传输一个修改的间接数据定址字的计数字段中指定的数据量，将对于数据传输的控制传递到列表中的另一个修改的间接数据定址字。

这里还描述并声明了对应于上文概括的方法的系统和计算机程序产品。

通过本发明的技术，来实现其他功能和优点。这里将详细描述本发明的其他实施例和方面，它们被视为所声明的发明的一部分。

附图说明

在本说明书结尾时在权利要求书中特别指出了并作为示例明确声明了本发明的一个或多个方面。通过下面的结合附图对本发明进行的详细说明，可以更好地理解本发明的前述和其他目的、特征和优点，其中：

图 1 描述了集成并使用了本发明的一个或多个方面的处理

环境的一个实施例；

图 2 描述了根据本发明的一个方面的与输入/输出处理关联的逻辑的一个示例；

图 3 描述了根据本发明的一个方面的与使用修改的间接数据定址字来控制数据传输关联的逻辑的一个实施例；以及

图 4 描述了集成了本发明的一个或多个方面的计算机程序产品的一个实施例。

具体实施方式

根据本发明的一个方面，在输入/输出 (I/O) 设备和存储器（例如，非连续的，即在物理上不是连续的主存储器、扩充存储器、主存储器、随机存取存储器，或可由中央处理单元直接访问的任何其他存储器）之间传输数据，不管存储器边界是什么。例如，数据可以在 2KB 或 4KB 字节的边界处传输。或者，使用修改的间接数据定址，以便在控制块的控制之下进行数据传输，这里称为修改的间接数据定址字 (MIDAW)。响应为一个 MIDAW 中的传输指定的数据量，并假设有更多数据要传输，控制被从一个 MIDAW 传递到另一个 MIDAW。控制的传递是独立于存储器边界（例如，2KB，4KB）进行的，从而在控制数据传输时具有灵活性。

这种灵活的数据传输控制可在许多环境中使用，包括但不限于，具有输入/输出子系统（如位于纽约 Armonk 的 IBM 公司提供的通道子系统）的处理环境。

将参考图 1 描述集成并使用了本发明的一个或多个方面的处理环境的一个示例。处理环境 100 包括，例如，主存储器 102、一个或多个中央处理单元 (CPU) 104、存储器控制元件 106、通道子系统 108、一个或多个控制单元 110 和一个或多个输入/输出 (I/O) 设备 112，下面将对它们中的每一个进行描述。

主存储器 102 存储了从输入设备 112 输入的数据和程序。主存储器 102 是直接可寻址的，并可以保证由中央处理单元 104 和通道

子系统 108 对数据进行高速的处理。在一个示例中，主存储器 102 包括顾客区域和系统区域（未显示）。

中央处理单元 104 是环境 100 的控制中心。它包含指令执行、中断操作、计时功能、初始程序加载及其他机器相关的功能的序列化和处理工具。中央处理单元 104 通过诸如双向的或单向总线之类的连接 114 而耦接到存储器控制元件 106。

存储器控制元件 106 通过诸如总线之类的连接 116 耦接到主存储器 102，通过连接 114 耦接到中央处理单元 104，通过连接 118（例如，双字总线）耦接到通道子系统 108。存储器控制元件 106 控制由 CPU 104 和通道子系统 108 作出的请求的队列化和执行。

如上文所描述的，通道子系统 108 耦接到存储器控制元件 106，并通过诸如串行链路之类的连接 120 而耦接到每一个控制单元。通道子系统 108 指示输入/输出设备 112 和主存储器 102 之间的信息流动。它免除了中央处理单元的直接与输入/输出设备进行通信的任务，并可使数据处理与输入/输出处理同时进行。在管理发往或发自输入/输出设备 112 的信息流时，通道子系统使用一个或多个通道路径 122 作为通信链路。作为输入/输出处理的一部分，通道子系统 108 还执行路径管理功能：进行通道路径可用性测试、选择可用通道路径和利用输入/输出设备启动操作的执行。

每一个通道路径 122 都包括通道 124（在一个示例中，通道位于通道子系统内，如图 1 所示）、一个或多个控制单元 110 和一个或多个连接 120。在另一个示例中，还可以具有一个或多个动态开关，作为通道路径的一部分。动态开关耦接到通道和控制单元，并提供在物理上互连连接到开关的任何两个链路的功能。

在通道子系统 108 内还有子通道（未显示）。为每一个输入/输出设备提供了一个子通道，并让该子通道专用于该输入/输出设备，所述输入/输出设备可以被程序通过通道子系统进行访问。子通道（例如，诸如表之类的数据结构）向程序提供了设备的逻辑外观。每一个子通道向通道子系统 108 提供关于关联的输入/输出设备 112 及其附

属装置的信息。子通道还提供关于输入/输出操作的信息及其他涉及关联的输入/输出设备的功能。子通道是通道子系统 108 用来向中央处理单元 104 提供有关关联的输入/输出设备 112 的信息,并通过执行输入/输出指令获取此信息的装置。

通道子系统 108 耦接到一个或多个控制单元 110。每一个控制单元都提供逻辑,以操作和控制一个或多个输入/输出设备并通过使用公用设施,使每一个输入/输出设备的特征适应由通道提供的链路接口。公用设施用于执行输入/输出操作,提供涉及输入/输出设备和控制单元的状态的指示,控制通过通道路径的数据传输的时间,以及输入/输出设备控制的某些级别。

每一个控制单元 110 都通过连接 126 (例如,总线)耦接到一个或多个输入/输出设备 112。输入/输出设备 112 接收信息或将信息存储在主存储器 102 中和/或其他存储器中。输入/输出设备的示例包括读卡器和穿孔器、磁带机、直接访问存储设备、显示器、键盘、打印机、指示设备、电传处理设备、通信控制器和基于传感器的设备等等。

通过执行指定与设备关联的子通道的输入/输出指令,利用设备 112 启动输入/输出操作。在一个实施例中,由使用通道命令字的 START SUBCHANNEL 指令来启动和控制输入/输出操作。START SUBCHANNEL 指令是由中央处理单元 104 执行的,是监控来自其他管理或处理输入/输出数据的程序的对输入/输出操作的请求的流的中央处理单元程序的一部分。

当执行 START SUBCHANNEL 指令时,参数被传递到请求通道子系统 108 利用与子通道关联的输入/输出设备执行开始功能的目标子通道。通道子系统通过用于子通道中的信息(包括在 START SUBCHANNEL 指令的执行过程中传递的信息)来查找到设备的可访问的通道路径,执行开始功能。一旦选择了设备,通过由通道子系统 108 和输入/输出设备 112 对通道命令字进行解码并执行,来实现输入/输出操作的执行。通道命令字指定要执行的

命令，为按顺序执行提供的一个或多个通道命令字构成了通道程序。作为一个示例，指令和通道命令字两者都是从主存储器 102 获取的。

在 2005 年 9 月出版的“IBM® z/Architecture Principles of Operation”，Publication No. SA22-7832-04（第 5 版）中详细描述了 START SUBCHANNEL 指令，以及通道，控制单元和通道命令字，在此引用了全部内容作为参考。在 1995 年 10 月 24 出版的授予 Cormier 等人的名称为“System For Transferring Data Between I/O Devices And Main Or Expanded Storage Under Dynamic Control Of Independent Indirect Address Words (IDAWS)”的美国专利 No. 5,461,721，以及 1996 年 6 月 11 日出版的授予 Casper 等人的名称为“Method And System For Pipelining The Processing Of Channel Command Words”的美国专利 No. 5,526,484 中也描述了这些组件中的一个或多个，这里引用了这些专利的全部内容作为参考。IBM 是位于美国纽约的 Armonk 的 IBM 公司的注册商标。这里所使用的其他名称可能是 IBM 公司或其他公司的注册商标、商标或产品名称。

下面将参考图 2 描述涉及处理传输数据的 I/O 请求的更多细节。如前所述，通过执行 START SUBCHANNEL 指令，启动 I/O 操作 (200)。START SUBCHANNEL 指令是由中央处理器启动的，并具有一个或多个字段，包括操作请求块 (ORB 202) 在存储器（例如，主存储器）中的地址。ORB 202 包括，根据本发明的一个方面，指定是否要使用修改的间接定址的控制 204，通道命令字 (CCW) 程序地址 206，用于定位要执行的通道程序的第一 CCW 208。当控制 204 被设置为例如，1 时，它指定通道程序可以包括 CCW，而 CCW 又指定 MIDAW 列表，这在下面将进一步地描述。

响应执行 START SUBCHANNEL 指令，开始执行通道程序。具体来说，处理程序的第一 CCW。CCW 具有许多控制处理的字段。这些字段包括，例如，指定要执行的操作的命令代码 210；用于控制 I/O 操作的多个标记 212；对于指定数据传输的命令，指定要传输的

由 CCW 指定的存储区中的字节数量的计数字段 214；以及数据地址 216，当不使用间接定址时，指向包括数据的主存储器中的位置，或在此情况下，当使用修改的间接数据定址时，指向要处理的修改的间接数据定址字 (MIDAW) 的列表（例如，连续的列表）。

有各种可以执行的 I/O 操作，包括但不限于，写入、读取、反向读、控制和检测操作。对于写入、读取、控制和检测操作，按照地址的升序来使用存储器位置。随着向存储器传输信息或从存储器传输信息，地址字段中的地址增大，计数字段中的计数减少。反向读操作按照地址的降序将数据放置在存储器中，计数和地址两者都减少。当计数到达零时，由 CCW 定义的存储区用完。

根据本发明的一个方面，使用修改的间接数据定址 (MIDA)。MIDA 允许单个通道命令字控制横跨主存储器的非连续的块（或在另一个实施例中，其他存储器），例如，最长达 65,535 字节的数据的传输。可以在任何边界上指定要传输的每一个存储器块，存储器块可以是任何长度。由 ORB 中的标记控制修改的间接数据定址，并由 CCW 中的标记指定修改的间接数据定址，当设置了两者时，表明 CCW 数据地址不用于直接对数据进行定址，而是指向一个或多个修改的间接数据定址字 (MIDAW) 的连续的列表。列表中的 MIDAW 的数量取决于 CCW 中的计数，因为所有 MIDAW 计数的总和等于 CCW 计数。将会有足够的 MIDAW 传输 CCW 计数字段中指定的数据量。

作为示例，由操作系统或设备驱动程序创建每一个 MIDAW 220，具有多个字段，例如包括：

标记 222：标记字段包括多个标记，如：

最后一个 MIDAW 标记，当设置时，指定此 MIDAW 是 MIDAW 的连续的列表中的最后一个；

跳过标记，当设置时，指定在读取、反向读、检测 ID，或检测操作过程中抑制到存储器的信息传输，如此指定跳过生效。当操作不是读取、反向读、检测 ID 或检测时，忽略此标记，跳过不生

效；以及

数据传输中断控制标记，当设置时，指定当设备试图传输数据时，识别程序检查。

计数 225: MIDAW 的计数字段指定由数据地址字段指定的存储区中的字节数量。当跳过生效时，作为一个示例，计数值可以在 1-65,535 的范围内。当计数值导致总的数据传输计数超过 CCW 计数字段中指定的值时，识别程序检查条件。

数据地址 226: MIDAW 的此字段指定主存储器中的位置的地址，作为示例，当 MIDAW 用于数据传输时，是要传输的信息的第一字节。如果计数字段指定零，则不检查此字段。

当指定修改的间接数据定址时，CCW 的数据地址字段指定用于 CCW 命令的数据传输的第一 MIDAW 的位置。如果完成 CCW 的数据传输需要的话，从存储器中的连续的位置获取另外的 MIDAW。由 CCW 的计数字段确定用于 CCW 的 MIDAW 的数量（相对于由 CCW 指定的 MIDAW 的列表中的计数字段）。

由 CCW 计数字段限制可以由单个 MIDAW 列表传输或跳过或既传输又跳过的字节的总数。由 CCW 指定的列表中的所有 MIDAW 中的计数字段的总和应该等于 CCW 的计数字段中的值。

由 CCW 指定的 MIDAW 可以指定存储器内的任何位置。当 MIDAW 跳过标记是零，CCW 指定读取、写入、控制、检测 ID 或检测命令时，从（或向）连续的较高的存储器位置传输数据，直到由 MIDAW 计数字段指定的字节数量都已经传输完毕。当 MIDAW 跳过标记是零，CCW 指定反向读命令时，向连续的较低的存储器位置传输数据，直到由 MIDAW 计数字段指定的字节数量都已经传输完毕。当 MIDAW 跳过标记是 1，CCW 指定读取、控制、检测 ID 或检测命令时，跳过连续的较高的存储器位置，直到由 MIDAW 计数字段指定的字节数量都已经跳过。当 MIDAW 跳过标记是 1，CCW 指定反向读命令时，跳过连续的较低的存储器位置，直到由 MIDAW 计数字段指定的字节数量都已经跳过。

当已经传输完或跳过指定的字节数量并指定随后的 MIDAW 时，将对于数据传输的控制传递到列表中的下一个 MIDAW。类似于 CCW 指定的 MIDAW，随后的 MIDAW 可以指定任何位置和任何长度。

虽然图 2 中描述了一个 MIDAW 和一个 CCW，但是，CCW 可以与 MIDAW 的列表关联，ORB 可以指向要处理的多个 CCW 中的一个 CCW。多个 CCW 中的每一个 CCW 可以指出也可以不指出一个或多个 MIDAW。

可以预取有关当前 CCW 或预取的（在使用前面获取的）CCW 的 MIDAW。在一个实施例中，可以预取的 MIDAW 的数量不超过满足指定 MIDAW 的 CCW 中的计数所需的数量。不使用为 CCW 预取的并且不接收对 I/O 操作的控制的任何 MIDAW。将控制从一个 MIDAW 转移到下一个 MIDAW 的操作对任何附属的设备来说是透明的。当起作用的 CCW 取得控制（对于列表中的第一 MIDAW），或当由前一 MIDAW 指定的最后一个字节已经传输完毕（对于列表中的所有随后的 MIDAW）并且起作用的 CCW 指定另外的数据的传输时，MIDAW 取得对于数据传输的控制。

如果 CCW 中的计数已经到达零，并传输了前一 MIDAW 的数据的最后一个字节，则 MIDAW 不取得对 I/O 操作的控制。不向程序指出在预取的 MIDAW 中检测到程序或访问错误，直到 MIDAW 取得对于数据传输的控制，即使作出了预取该数据的尝试。

下面将参考图 3 描述有关处理 MIDAW 的更多细节。首先，构建通道程序，步骤 300。通道程序包括一个或多个用于控制数据传输的 CCW。此后，发出 START SUBCHANNEL 指令，步骤 302，并获取和使用第一 CCW，步骤 304。ORB 中的标记和 CCW 中的标记表明将使用 MIDAW，如此，获取并使用第一 MIDAW，步骤 306。不管存储器边界如何，传输 MIDAW 中的指定的数据量，步骤 308。此后，就是否有更多数据要传输作出判断，查询 310。如果有更多数据要传输，如 CCW 中的计数字段所示，那么，就是否有另一个

MIDAW 要选择作出判断, 查询 312。如果当前 MIDAW 中的标记表明这是最后一个 MIDAW, 那么, 指出一个错误, 表明没有足够的 MIDAW 传输请求的数据量, 步骤 314。否则, 选择另一个 MIDAW, 步骤 306。

返回到 INQUIRY 310, 如果对于此 CCW 没有更多数据需要传输, 就是否存在另一个 MIDAW 待处理作出进一步的查询, 查询 315。如果是, 表明不匹配, 表明比需要传输的数据有更多的 MIDAW, 步骤 316。然而, 如果没有另一个 MIDAW 要处理 (或者, 在另一个实施例中, 在指定错误 314 和/或错误 316 之后), 就是否有更多 CCW 要处理作出判断, 查询 318。在一个示例中, 这是通过当前 CCW 中的标记来判断的。如果有更多 CCW 要执行, 则继续执行步骤 304。否则, 对通道程序的处理完成。

上文详细描述了在修改的间接数据定址字的控制之下, 灵活地控制例如, I/O 设备和非连续的存储器之间的数据传输的技术。修改的间接数据定址提供了在设备和多个非连续的存储器块之间传输数据的更加灵活的并且更加有用的技术。它允许单个 CCW 控制设备和映射到虚拟地址空间中的页面的存储器之间的任何边界 (即, 非分页边界) 上的数据传输的开始和结束。MIDA 还允许传输大于非连续的 4KB 块的数据。

本发明的一个或多个方面可以包括在具有计算机可使用的介质的产品中 (例如, 一个或多个计算机程序产品和/或系统)。例如, 介质中包含了用于提供和实现本发明的功能的计算机可读程序代码装置或逻辑 (例如, 指令、代码、命令等等)。产品可以作为计算机系统的一部分或单独地销售。

下面将参考图 4 描述集成了本发明的一个或多个方面的计算机程序产品的一个示例。计算机程序产品 400 包括, 一个或多个计算机可使用的介质 402, 在其上存储了计算机可读程序代码装置或逻辑 404, 以提供和实现本发明的一个或多个方面。介质可以是电子的、磁性的、光学的、电磁的、红外线的或半导体系统 (或设备) 或传播介

质。计算机可读介质的示例包括半导体或固态存储器，磁带，可移动计算机磁盘，随机存取存储器 (RAM)，只读存储器 (ROM)，硬磁盘和光盘。光盘的示例包括光盘-只读存储器 (CD-ROM)，光盘读取/写入 (CD-R/W) 和 DVD。

程序指令的序列或由一个或多个计算机可读程序代码装置或逻辑定义的一个或多个相互关联的模块指示了本发明的一个或多个方面的性能。

优选情况下，提供了用于传输数据的扩散/收集技术。提供了其中要传输的数据量独立于指定存储器边界的传输数据功能。此技术消除了其他间接数据定址技术施加的程序可选择的最大 2KB/4KB 块大小的限制。此外，单独的传输计数与每一个间接数据定址关联，基于与间接数据定址关联的计数字段，将控制从一个列出的间接数据定址转移到下一个地址。还允许根据每个间接地址和关联的计数而实施跳越，并对间接数据定址的列表的大小进行验证。

优选情况下，可以在任何边界和长度上（例如，在一个实施例中，直到 64K-1）指定不连续的存储器块，而不使用链接的数据 CCW。

修改的间接数据定址提供了这样的机制：程序可以指定通道程序，该通道程序控制横跨非连续的存储器块（位于任何边界和任何大小）的数据传输。这样的通道程序包括，例如，一个或多个通道命令字，其地址字段不用于对数据定址，而被用来指定修改的间接数据定址字的连续的列表的地址。MIDAW 的列表不是地址列表，而是控制块的列表，其中，每一个控制块（即，MIDAW）包括在处理中使用的信息。

每一个 MIDAW 都包括计数字段，该计数字段指定当该 MIDAW 处于控制下时要传输的数据量。将控制从列表中的一个 MIDAW 转移到下一个的过程是由 CCW 计数字段和 MIDAW 计数字段的组合控制的。当数据传输开始时，列表中的第一 MIDAW 被用来指定，例如传输在主存储器中的什么位置开始。如果 CCW 计数字段的值超过第一 MIDAW 中的计数字段的值，并且最后一个

MIDAW 标记是零，则下一个 MIDAW 承担控制，数据传输在由 MIDAW 指定的主存储器地址处继续。这一将控制传递到列表中的连续的 MIDAW 的过程持续进行，直到已经传输了由 CCW 计数字段指定的数据的总量。

由于 MIDAW 包括地址字段和计数字段，由任何 MIDAW 指定的存储器位置可以在任何边界上开始和结束。

此外，每一个 MIDAW 都包括其中有跳过指示器的标记字段。如果跳过指示器是当 MIDAW 在数据传输过程中获取控制时的指示器，则跳过由 MIDAW 指定的存储器的范围，对于该范围，不进行数据传输。

MIDAW 标记字段还包括 MIDAW 是列表中的最后一个 MIDAW 的指示器。如此，如果当此指示器是 1 时 CCW 计数字段指定将控制转移到随后的 MIDAW，则在 CCW 和 MIDAW 列表之间存在不匹配。类似地，如果已经传输了由 CCW 指定的数据总量，当前控制中的 MIDAW 将其标记设置为 0，则在 CCW 和 MIDAW 列表之间也有不匹配。不论是哪一种情况，都可以通过通道程序检查，通知该程序，以指出错误。

虽然上文描述了各种实施例，但是，这些只是示例而已。这里所描述的处理环境之外的处理环境，包括使用通道子系统之外的 I/O 子系统的其它处理环境，也可以集成和使用本发明的一个或多个方面。此外，虽然显示了各种控制块，但是，信息在这些控制块内的位置也可以是这里所示的之外的情况。此外，每一个控制块都可以包括比这里所描述的情况更多，稍少的，或不同的信息。例如，可以有更多的，较少的和/或不同的字段，包括那些可以包括更多的，较少的和/或不同的标记的字段。此外，还可以有更多的，较少的和/或不同的字段大小。此外，虽然在实施例的各个部分提及或描述了主存储器，但是，本发明的一个或多个方面也可以适用于其他存储器。

在另外的实施例中，如果需要，也可以实现某些限制。例如，可以实现不能指定交叉特定边界（如 4KB 字节边界）的 MIDAW 列

表的限制。这会将最大 MIDAW 列表大小限制到 256 MIDAW。此外，单个 MIDAW 还可以指定可以不超过特定边界（如 4KB 边界）的数据计数，和/或不指定跨 4KB 字节边界的数据传输。在其他的示例中，跳过标记不必只应用于读取操作。在不偏离本发明的精神的情况下，可以进行任何其他更改和/或增强。

此外，环境可以包括仿真器（例如，软件或其他模拟机制），其中，模拟了特定体系结构或其子集。在这样的环境中，仿真器的一个或多个模拟功能可以实现本发明的一个或多个方面，尽管执行仿真器的计算机可能与正在被模拟的功能具有不同的体系结构。作为一个示例，在仿真模式下，对正在被模拟的特定指令或操作进行解码，构建适当的模拟功能，以实现单个指令或操作。

在模拟环境中，主机计算机包括，例如，用于存储指令和数据的存储器；指令获取单元，用于从存储器中获取指令，可选地，为获取的指令提供本地缓冲区；指令解码单元，用于接收指令获取单元，判断已经获取的指令的类型；以及用于执行指令的指令执行单元。执行可以包括将数据加载到存储器的寄存器中；从寄存器中将数据存储回存储器中；或执行某种类型的算术或逻辑运算，如由解码单元所判断的。在一个示例中，每一个单元都是以软件实现的。例如，由单元执行的操作是作为仿真器软件内的一个或多个子例程实现的。

此外，可使用适用于存储和/或执行程序代码的数据处理系统，该系统包括至少一个通过系统总线直接或间接地耦接到存储器元件的处理器。存储器元件包括，例如，在程序代码的实际执行过程中使用的本地存储器，大容量存储器，以及高速缓冲存储器，该高速缓冲存储器提供了至少某些程序代码的临时存储器，以便减少在执行过程中必须从大容量存储器检索代码的次数。

输入/输出或 I/O 设备（包括但不限于，键盘、显示器、指示设备，等等）可以直接或者通过 I/O 控制器耦接到系统。网络适配器也可以耦接到系统，以使得数据处理系统通过专用或公共网络耦接到其他数据处理系统或远程打印机或存储设备。调制解调器，电缆调制

解调器和以太网卡只是几个可用的网络适配器类型。

本发明的一个或多个方面的功能可以以软件、固件、硬件或其某些组合来实现。可以提供至少一个机器可读程序存储设备，其中包含了由机器可执行的至少一个指令程序以执行本发明的功能。

这里所描述的流程图只是示例。在不偏离本发明的精神的情况下，对这里所描述的这些图表或步骤（或操作）可以有許多更改。例如，可以以不同顺序执行各个步骤，也可以添加、删除或修改步骤。所有这些变化都被视为权利要求所要求保护的本发明的一部分。

虽然这里详细描述了优选实施例，但是，对于那些本领域技术人员来说显而易见的是，在不偏离本发明的精神的情况下，可以进行各种修改、添加，替换等等，因此，也被视为在如下面的权利要求所定义的本发明的范围内。

图1

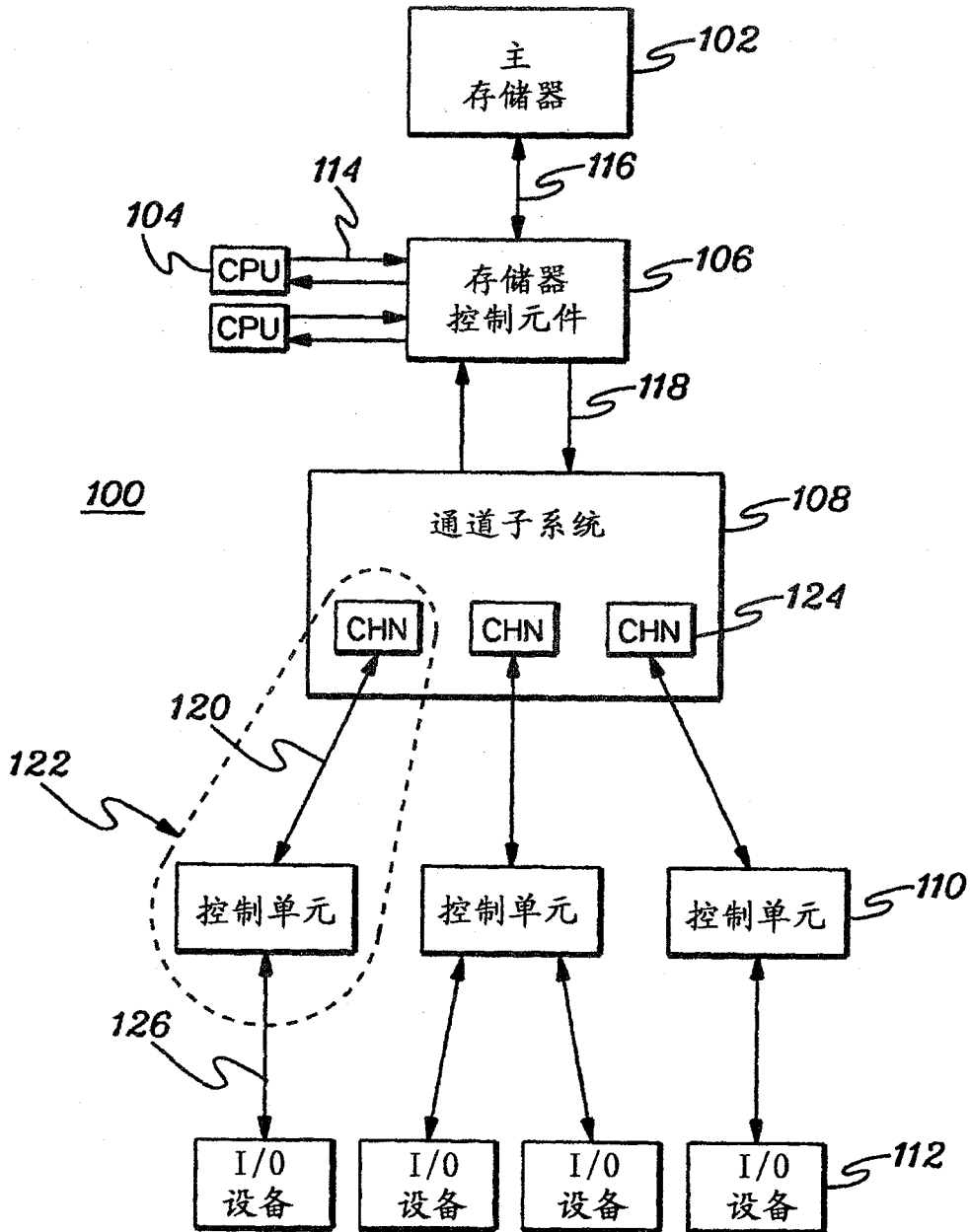


图 2

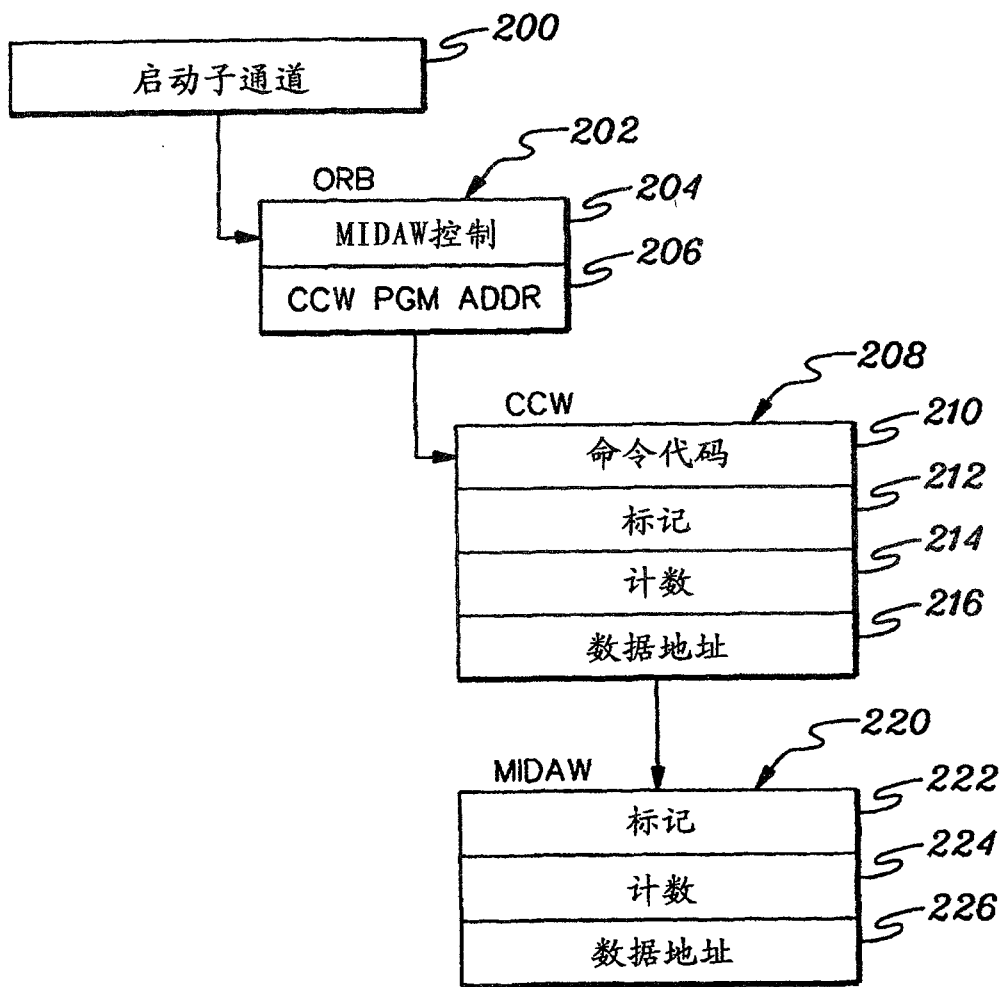


图 3

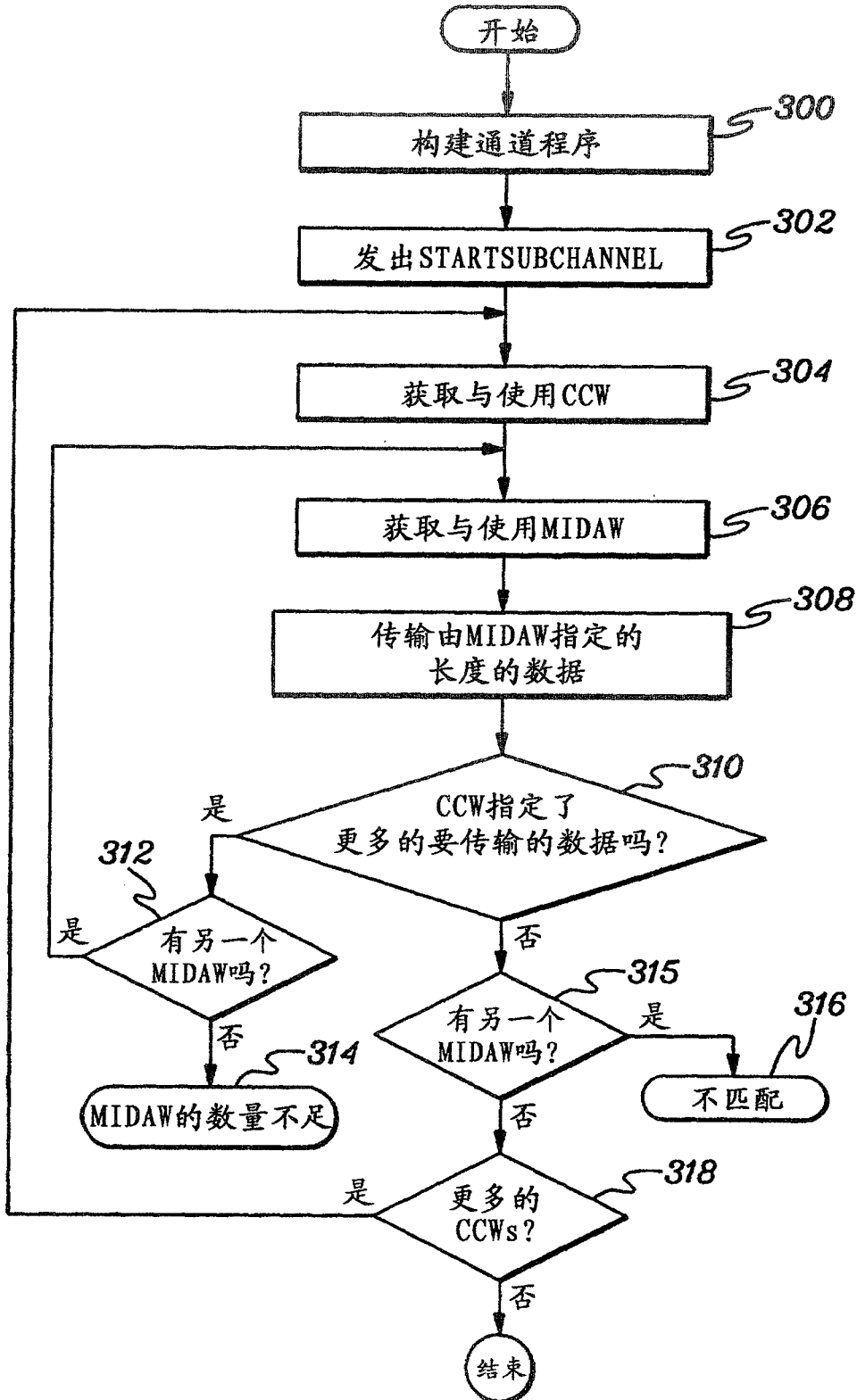


图4

