



(12) 发明专利申请

(10) 申请公布号 CN 103582870 A

(43) 申请公布日 2014. 02. 12

(21) 申请号 201280026402. 8

(51) Int. Cl.

(22) 申请日 2012. 05. 10

G06F 9/30(2006. 01)

G06F 12/02(2006. 01)

(30) 优先权数据

13/157, 710 2011. 06. 10 US

(85) PCT国际申请进入国家阶段日

2013. 11. 29

(86) PCT国际申请的申请数据

PCT/EP2012/058674 2012. 05. 10

(87) PCT国际申请的公布数据

W02012/168027 EN 2012. 12. 13

(71) 申请人 国际商业机器公司

地址 美国纽约

(72) 发明人 P · K · 什韦德 K · J · 奥克斯

P · G · 苏顿 P · D · 德里费尔

H · 尤登弗兰德 S · G · 格拉森

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 张亚非 于静

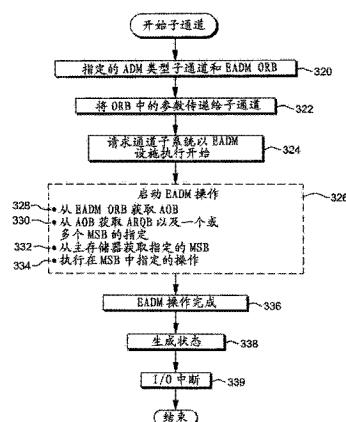
权利要求书4页 说明书64页 附图24页

(54) 发明名称

清除存储级内存的块

(57) 摘要

提供了用于存储级内存的抽象，其向程序隐藏了存储级内存的实现细节，且提供了用于执行诸如控制主存储器和存储级内存之间的数据移动或管理存储级内存等特定动作的标准通道编程接口。



1. 一种用于在包括主存储器和存储级内存的计算环境中执行开始子通道指令的计算机程序产品，所述计算机程序产品包括：

计算机可读存储媒介，其可被处理电路读取并存储用于由所述处理电路执行以执行方法的指令，所述方法包括：

响应于确定由所述开始子通道指令指示的子通道是异步数据移动器(ADM)子通道，执行：

从主存储器中获得操作请求块，所述操作请求块包括操作块的地址；

基于所述操作块的所述地址，从主存储器中获得所述操作块，所述操作块包括：请求块、响应块、和一个或多个移动规格块(MSB)，其中所述请求块包括具有指示被包括在所述操作块中并且被所述操作块引用的一个或多个 MSB 的数目的值的 MSB 计数字段，其中所述响应块被配置为保持异常条件，其中每个移动规格块被配置为包括操作代码字段、块计数字段、用于寻址主存储器的主存储器地址字段、用于寻址存储级内存的存储级内存地址字段、块大小字段和标志字段；

针对一个或多个所述 MSB 中的每个 MSB 执行方法，所述方法包括：

获得所述一个或多个移动规格块中的一个移动规格块；

基于具有第一值的所述获得的 MSB 的标志字段，执行方法，该方法包括：

基于所述获得的移动规格块中的所述操作代码，确定所述获得的移动规格块被配置为清除存储级内存的块；

基于所述获得的移动规格块的所述块大小字段和所述块计数字段确定将被清除的存储级内存的所述块的大小和将被清除的块的数目；并且

基于所述获得移动规格块，清除与所述获得的 MSB 的所述存储级内存地址字段的所述存储级内存地址相关联的存储级内存的块。

2. 根据权利要求 1 所述的计算机程序产品，其中所述方法还包括：

基于具有第二值的所述获得的 MSB 的标志字段，执行方法，该方法包括：

使用所述获得的 MSB 的主存储器地址作为地址来获得另一个移动规格块；

基于所述另一个 MSB 的操作代码字段，确定所述另一个移动规格块是否被配置为从主存储器中清除存储级内存的块，其中存储块的大小由所述另一个 MSB 的块大小字段来确定并且清除的块的数目由所述另一个 MSB 的块计数字段来确定；并且

清除与所述另一个 MSB 的存储级内存地址字段的存储级内存地址相关联的存储级内存的块。

3. 根据权利要求 2 所述的计算机程序产品，其中所述清除存储块包括将所述存储块设置为零。

4. 根据权利要求 1 所述的计算机程序产品，其中所述清除存储块包括将所述存储块设置为零。

5. 根据权利要求 1 所述的计算机程序产品，其中响应于所述获得的移动规格块的操作代码指示清除操作，所述主存储器地址字段没有意义。

6. 根据权利要求 1 所述的计算机程序产品，其中在所述块大小字段中所指定的大小是 4K 或者 1M。

7. 根据权利要求 1 所述的计算机程序产品，其中所述方法还包括响应于所述清除的完

成,生成一个或多个状态条件,所述一个或多个状态条件对于所述开始子通道指令的发出者是可访问的。

8. 根据权利要求 1 所述的计算机程序产品,其中响应于在所述清除操作的执行期间检测到错误条件所述响应块被填充。

9. 一种用于在包括主存储器和存储级内存的计算环境中执行开始子通道指令的计算机系统,所述计算机系统包括 :

存储器 ;和

与所述存储器通信的处理器,其中所述计算机系统被配置为执行方法,所述方法包括 :

响应于确定由所述开始子通道指令指示的子通道是异步数据移动器(ADM) 子通道,执行 :

从主存储器中获得操作请求块,所述操作请求块包括操作块的地址 ;

基于所述操作块的所述地址,从主存储器中获得所述操作块,所述操作块包括 :请求块、响应块、和一个或多个移动规格块(MSB),其中所述请求块包括具有指示被包括在所述操作块中并且被所述操作块引用的一个或多个 MSB 的数目的值的 MSB 计数字段,其中所述响应块被配置为保持异常条件,其中每个移动规格块被配置为包括操作代码字段、块计数字段、用于寻址主存储器的主存储器地址字段、用于寻址存储级内存的存储级内存地址字段、块大小字段和标志字段 ;

针对一个或多个所述 MSB 中的每个 MSB 执行方法,所述方法包括 :

获得所述一个或多个移动规格块中的一个移动规格块 ;

基于具有第一值的所述获得的 MSB 的标志字段,执行方法,该方法包括 :

基于在所述获得的移动规格块中的所述操作代码,确定所述获得的移动规格块被配置为清除存储级内存的块 ;

基于所述获得的移动规格块的所述块大小字段和所述块计数字段确定将被清除的存储级内存的所述块的大小和将被清除的块的数目 ;并且

基于所述获得移动规格块,清除与所述获得的 MSB 的所述存储级内存地址字段的所述存储级内存地址相关联的存储级内存的块。

10. 根据权利要求 9 所述的计算机系统,其中所述方法还包括 :

基于具有第二值的所述获得的 MSB 的标志字段,执行方法,该方法包括 :

使用所述获得的 MSB 的主存储器地址作为地址来获得另一个移动规格块 ;

基于所述另一个 MSB 的操作代码字段,确定所述另一个移动规格块是否被配置为从主存储器中清除存储级内存的块,其中存储块的大小由所述另一个 MSB 的块大小字段来确定并且清除的块的数目由所述另一个 MSB 的块计数字段来确定 ;并且

清除与所述另一个 MSB 的存储级内存地址字段的存储级内存地址相关联的存储级内存的所述块。

11. 根据权利要求 10 所述的计算机系统,其中所述清除存储级块包括将所述存储块设置为零。

12. 根据权利要求 9 所述的计算机系统,其中所述清除存储块包括将所述存储块设置为零。

13. 根据权利要求 9 所述的计算机系统,其中响应于所述获得的移动规格块的操作代码指示清除操作,主存储器的地址字段没有意义。

14. 根据权利要求 9 所述的计算机系统,其中在所述块大小字段中所指定的大小是 4K 或者 1M。

15. 根据权利要求 9 所述的计算机系统,其中所述方法还包括响应于所述清除的完成,生成一个或多个状态条件,所述一个或多个状态条件对于所述开始子通道指令的发出者是可访问的。

16. 根据权利要求 9 所述的计算机系统,其中响应于在所述清除操作的执行期间检测到错误条件所述响应块被填充。

17. 一种用于在包括主存储器和存储级内存的计算环境中执行开始子通道指令的方法,所述方法包括:

 响应于确定由所述开始子通道指令指示的子通道是异步数据移动器(ADM) 子通道,执行:

 从主存储器中获得操作请求块,所述操作请求块包括操作块的地址;

 基于所述操作块的所述地址,从主存储器中获得所述操作块,所述操作块包括:请求块、响应块、和一个或多个移动规格块(MSB),其中所述请求块包括具有指示被包括在所述操作块中并且被所述操作块引用的一个或多个 MSB 的数目的值的 MSB 计数字段,其中所述响应块被配置为保持异常条件,其中每个移动规格块被配置为包括操作代码字段、块计数字段、用于寻址主存储器的主存储器地址字段、用于寻址存储级内存的存储级内存地址字段、块大小字段和标志字段;

 针对一个或多个所述 MSB 中的每个 MSB 执行方法,所述方法包括:

 获得所述一个或多个移动规格块中的一个移动规格块;

 基于具有第一值的所述获得的 MSB 的标志字段,执行方法,该方法包括:

 基于所述获得的移动规格块中的所述操作代码,确定所述获得的移动规格块被配置为清除存储级内存的块;

 基于所述获得的移动规格块的所述块大小字段和所述块计数字段确定将被清除的存储级内存的所述块的大小和将被清除的块的数目;并且

 基于所述获得移动规格块,清除与所述获得的 MSB 的所述存储级内存地址字段的所述存储级内存地址相关联的存储级内存的块。

18. 根据权利要求 17 所述的方法,还包括:

 基于具有第二值的所述获得的 MSB 的标志字段,执行方法,该方法包括:

 使用所述获得的 MSB 的主存储器地址作为地址来获得另一个移动规格块;

 基于所述另一个 MSB 的操作代码字段,确定所述另一个移动规格块是否被配置为从主存储器中清除存储级内存的块,其中存储块的大小由所述另一个 MSB 的块大小字段来确定并且清除的块的数目由所述另一个 MSB 的块计数字段来确定;并且

 清除与所述另一个 MSB 的存储级内存地址字段的存储级内存地址相关联的存储级内存的块。

19. 根据权利要求 17 所述的方法,还包括响应于所述清除的完成,生成一个或多个状态条件,所述一个或多个状态条件对于所述开始子通道指令的发出者是可访问的。

20. 根据权利要求 17 所述的方法, 其中响应于在所述清除操作的执行期间检测到错误条件所述响应块被填充。

清除存储级内存的块

技术领域

[0001] 本发明的一个或多个方面一般涉及计算环境的辅助存储器，特别涉及辅助存储器的管理方面。

背景技术

[0002] 计算环境可包括主存储器(也称为主内存)，以及辅助存储器。主存储器是可被处理器访问的存储器，其可例如由绝对地址随机寻址。与诸如直接存取存储设备(DASD)或存储级内存的辅助存储器相比，主存储器被认为是快速存取存储器，而且，主存储器的寻址被认为是比 DASD 或存储级内存的寻址更简单。

[0003] 存储级内存，其是经典主存储器外的外部存储空间，提供了比直接存取存储设备更快的存取。与 DASD 不同，存储级内存典型地没有被实施为机械臂旋转盘，而是非机械地固态部件。典型地，存储级内存被实施为通过若干输入 / 输出(I/O)适配器连接到计算系统的固态设备组，所述输入 / 输出适配器被用来将 I/O 设备的技术映射到中央处理单元的存储总线。

发明内容

[0004] 通过提供用于在包括主存储器和存储级内存的计算环境中执行开始子通道指令的计算机程序产品，现有技术的缺点被克服并且本发明的优点被提供。所述计算机程序产品包括计算机可读的存储媒介，其可被处理电路读取并存储用于由所述处理电路执行以执行本发明方法的指令。所述方法包括，例如，响应于确定由所述开始子通道指令指示的子通道是异步数据移动器(ADM)子通道，执行：从主存储器中获得操作请求块，所述操作请求块包括操作块的地址；基于所述操作块的所述地址，从主存储器中获得所述操作块，所述操作块包括请求块、响应块、和一个或多个移动规格块(MSB)，其中所述请求块包括具有指示被包括在所述操作块中并且被所述操作块引用的一个或多个 MSB 的数目的值的 MSB 计数字段，其中所述响应块被配置为保持异常条件，其中每个移动规格块被配置为包括操作代码字段、块计数字段、用于寻址主存储器的主存储器地址字段、用于寻址存储级内存的存储级内存地址字段、块大小字段和标志字段；针对一个或多个所述 MSB 中的每个 MSB 执行方法，该方法包括：获得所述一个或多个移动规格块中的一个移动规格块；基于具有第一值的所述获得的 MSB 的所述标志字段，执行方法，该方法包括：基于所述获得的移动规格块中的操作代码，确定所述获得的移动规格块被配置为清除存储级内存的块；基于所述获得的移动规格块的所述块大小字段和所述块计数字段确定将被清除的存储级内存的所述块的大小和将被清除的块的数目；并且基于所述获得的移动规格块，清除与所述获得的 MSB 的所述存储级内存地址字段的所述存储级内存地址相关联的存储级内存的块。

[0005] 涉及本发明的一个或多个方面的方法和系统也在此描述并要求保护。而且，与本发明的一个或多个方面有关的服务也在此描述并可要求保护。

[0006] 额外的特点和优势通过本发明的技术被实现。本发明的其他实施例和方面在此被

详细描述并被认为是所要求保护的发明的一部分。

附图说明

[0007] 在本说明书的结尾处,本发明的一个或多个方面作为示例被特别指出并且明确要求保护。从以下结合附图的详细描述中本发明的上述和其他目的、特征和优点将变得明显,在附图中:

- [0008] 图 1A 示出了包含并使用本发明的一个或多个方面的计算环境的一个实施例;
- [0009] 图 1B 示出了包含并使用本发明的一个或多个方面的计算环境的另一个实施例;
- [0010] 图 2A 示出了根据本发明的一个方面使用的扩展异步数据移动器操作请求块的一个实施例;
- [0011] 图 2B 示出了根据本发明的一个方面使用的扩展异步数据移动器操作块的一个实施例;
- [0012] 图 2C 示出了根据本发明的一个方面使用的扩展异步数据移动器请求块的一个实施例;
- [0013] 图 2D 示出了根据本发明的一个方面使用的扩展异步数据移动器响应块的一个实施例;
- [0014] 图 2E 示出了根据本发明的一个方面使用的扩展异步数据移动器移动规格块的一个实施例;
- [0015] 图 2F 示出了根据本发明的一个方面使用的扩展异步数据移动器间接数据地址字的一个实施例;
- [0016] 图 3A 示出了根据本发明的一个方面使用的开始子通道指令的一个实施例;
- [0017] 图 3B 示出了根据本发明的一个方面的与开始子通道指令有关的逻辑的一个实施例;
- [0018] 图 3C 示出了根据本发明的一个方面使用的清除子通道指令的一个实施例;
- [0019] 图 3D 示出了根据本发明的一个方面使用的测试子通道指令的一个实施例;
- [0020] 图 3E 示出了根据本发明的一个方面使用的修改子通道指令的一个实施例;
- [0021] 图 3F 示出了根据本发明的一个方面使用的存储子通道指令的一个实施例;
- [0022] 图 4A 示出了根据本发明的一个方面使用的扩展异步数据移动器子通道信息块的一个实施例;
- [0023] 图 4B 示出了根据本发明的一个方面使用的扩展异步数据移动器路径管理控制字的一个实施例;
- [0024] 图 4C 示出了根据本发明的一个方面使用的子通道状态字的一个实施例;
- [0025] 图 4D 示出了根据本发明的一个方面使用的扩展异步数据移动器扩展状态字的一个实施例;
- [0026] 图 5A 示出了根据本发明的一个方面使用的存储存储级内存信息请求块的一个实施例;
- [0027] 图 5B 示出了根据本发明的一个方面使用的存储存储级内存信息响应块的一个实施例;
- [0028] 图 5C 示出了根据本发明的一个方面使用的存储级内存地址列表项的一个实施

例；

[0029] 图 5D 示出了根据本发明的一个方面的与存储存储级内存信息命令有关的逻辑的一个实施例；

[0030] 图 6A 示出了根据本发明的一个方面表示存储级内存的各个状态的状态图的一个例子；

[0031] 图 6B 示出了根据本发明的一个方面示出操作状态和数据状态的状态图的一个实施例；

[0032] 图 7A 示出了根据本发明的一个方面使用的配置存储级内存请求块的一个实施例；

[0033] 图 7B 示出了根据本发明的一个方面使用的配置存储级内存响应块的一个实施例；

[0034] 图 7C-7D 示出了根据本发明的一个方面使用的与配置存储级内存命令有关的逻辑的一个实施例；

[0035] 图 8A 示出了根据本发明的一个方面使用的通知响应块的一个例子；

[0036] 图 8B 示出了根据本发明的一个方面使用的存储事件信息请求块的一个实施例；

[0037] 图 8C 示出了根据本发明的一个方面使用的存储事件信息响应块的一个实施例；

[0038] 图 9A 示出了根据本发明的一个方面使用的解除配置存储级内存请求块的一个实施例；

[0039] 图 9B 示出了根据本发明的一个方面使用的存储级内存增量请求列表项的一个实施例；

[0040] 图 9C 示出了根据本发明的一个方面使用的解除配置存储级内存响应块的一个实施例；

[0041] 图 9D-9E 示出了根据本发明的一个方面使用的与解除配置存储级内存命令有关的逻辑的一个实施例；

[0042] 图 10 示出了包含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；

[0043] 图 11 示出了包含并使用本发明的一个或多个方面的主机计算机系统的一个实施例；

[0044] 图 12 示出了包含并使用本发明的一个或多个方面的计算机系统的又一个例子；

[0045] 图 13 示出了包括包含并使用本发明的一个或多个方面的计算机网络的计算机系统的另一个例子；

[0046] 图 14 示出了包含并使用本发明的一个或多个方面的计算机系统的各种组件的一个实施例；

[0047] 图 15A 示出了包含并使用本发明的一个或多个方面的图 14 的计算机系统的执行单元的一个实施例；

[0048] 图 15B 示出了包含并使用本发明的一个或多个方面的图 14 的计算机系统的分支单元的一个实施例；

[0049] 图 15C 示出了包含并使用本发明的一个或多个方面的图 14 的计算机系统的加载 / 存储单元的一个实施例；以及

[0050] 图 16 示出了包含并使用本发明的一个或多个方面的仿真的主机计算机系统的一

个实施例。

具体实施方式

[0051] 根据一个或多个实施例，提供了一种用于存储级内存的抽象，其对程序(例如操作系统)隐藏了存储级内存的实施细节，并提供了标准的通道编程接口以执行特定动作，诸如控制主存储器和存储级内存之间的数据移动。

[0052] 在一个例子中，提供了被称为扩展异步数据移动器(EADM)设施的设施，其允许程序请求主存储器和存储级内存之间的数据块的传送，以及请求其他操作。关于该设施的方面将在以下详细描述。

[0053] 而且，与 EADM 设施有关的是被称为 EAMD 释放设施的可选设施。当被安装时，EADM 释放设施提供了一种手段，用于由程序指定其不再需要求在一个或多个存储级内存块中保留数据。释放操作是否被支持用于所有的存储级内存块或仅用于存储级内存的一个子集是取决于模型的。

[0054] 一旦一个存储级内存块已被释放，程序可将新数据传递到块中，否则随后将数据从该块传送到主存储器的请求将会失败。

[0055] 将参考图 1A 描述包含并使用本发明的一个或多个方面的计算环境的一个实施例。计算环境 100 是基于例如位于纽约州阿蒙克市的国际商业机器公司(**IBM®**)提供的**z/Architecture®**。**z/Architecture®**的一个例子在 2010 年 8 月的 IBM 出版号 SA22-7832-08 的标题为“*z/Architecture Principles of Operation*”的**IBM®**出版物中被描述，该出版物的全部内容通过引用结合于此。在一个例子中，一种基于**z/Architecture®**的计算环境包括由国际商业机器公司提供的 zEnterprise196(z196)系统。**IBM®**和**z/Architecture®**以及 zEnterprise196 和 z196 是位于美国纽约州阿蒙克市的国际商业机器公司的注册商标。在此使用的其他名称也可能是国际商业机器公司或其他公司的注册商标、商标或产品名。

[0056] 作为一个例子，计算环境 100 包括经由一条或多条总线 106 连接到主存储器 104 的一个或多个中央处理单元 102。一个或多个中央处理单元可执行操作系统 108，诸如由国际商业机器公司提供的**z/OS®**操作系统。在其他例子中，一个或多个中央处理单元可执行其他的操作系统或不执行操作系统。**z/OS®**是位于美国纽约州阿蒙克市的国际商业机器公司的注册商标。

[0057] 中央处理单元 102 和主存储器 104 还可经由一个或多个连接 122(例如总线或其他连接)耦合到 I/O 集线器 120。I/O 集线器提供到一个或多个 I/O 适配器 130 的连接，所述 I/O 适配器又耦合到一个或多个固态设备 140。适配器和固态设备是存储级内存(例如闪速存储器)的实现。I/O 集线器是促进实施例的一个或多个方面的 I/O 子系统 145 的一部分。

[0058] 在又一个实施例中，如图 1B 所示，计算环境 150 可包括中央处理复合体(CPC)152，其基于由国际商业机器公司提供的**z/Architecture®**。中央处理复合体 152 包括例如一个或多个分区 154、管理程序 156、一个或多个中央处理器 158、以及输入 / 输出子系统 160

的一个或多个组件。在该例子中，分区 154 是逻辑分区（例如 LPAR），其每个包括被虚拟化为单独系统的一组系统硬件资源。

[0059] 每个逻辑分区 154 能用作单独系统。即，每个逻辑分区可被独立重置，如需要，初始加载操作系统或其他控制代码，并用不同的程序操作。在逻辑分区中运行的操作系统或应用程序看起来能访问完整和完全的系统，但是实际上，仅其一部分是可用的。硬件和许可内部代码（LIC）的组合，被称为固件，使一个逻辑分区中的程序避免受到不同逻辑分区中的程序干扰。这允许几个不同的逻辑分区以时间片的方式在单个或多个物理处理器上操作。如在此使用的，固件包括例如处理器（或执行处理的实体）的微代码、毫代码和 / 或宏代码。它包括例如在实现更高级的机器代码中使用的硬件级的指令和 / 或数据结构。在一个实施例中，它包括例如专有代码，其典型地被输送为微代码，该微代码包括可信软件或特定于底层硬件的微代码，并控制对系统硬件的操作系统访问。

[0060] 在该例子中，几个逻辑分区具有驻留操作系统（OS）170，其可于一个或多个逻辑分区不同。在一个实施例中，至少一个逻辑分区正在执行由纽约州阿蒙克市的国际商业机器公司提供的 **z/OS®** 操作系统。

[0061] 逻辑分区 154 由管理程序 156 管理，所述管理程序被在中央处理器 158 上运行的固件实施。逻辑分区 154 和管理程序 156 每个都包括驻留在与中央处理器关联的主存储器 159 的相应部分中的一个或多个程序。管理程序 156 的一个例子是由纽约州阿蒙克市的国际商业机器公司提供的处理器资源 / 系统管理器（RP/SM™）。

[0062] 中央处理器 158 是被分配给逻辑分区的物理处理器资源。例如，逻辑分区 154 包括一个或多个逻辑处理器，其每个表示被分配给该分区的物理处理器资源 158 的全部或一部分。特定分区 154 的逻辑处理器可以专用于该分区，这样底层处理器资源被保留用于该分区；或与其他分区共享，这样底层处理器资源潜在地可被另一个分区访问。

[0063] 输入 / 输出子系统 160（仅描述其一部分）提供了到存储级内存 180 的连接。在该例子中，地址空间被提供给存储级内存，该地址空间呈现内存是平坦（flat）的，对程序隐藏了物理实现的细节。作为一个例子，存在用于存储级内存的整个系统的一个地址空间，但从配置的角度（例如 LPAR 或在另一个实施例中是虚拟化客户（guest）），存在如下这样的地址空间，在其中为系统的每个配置填充存储级内存增量。存储级内存地址空间是单独的，且与主存储器地址空间分开。

[0064] 在 **z/Architecture®** 的一个特定例子中，I/O 子系统包括通道子系统，其不仅指引 I/O 外围控制单元（以及设备）和主存储器之间的信息流，也指引存储级内存和主存储器之间的信息流。但是，I/O 子系统也可以不是通道子系统。

[0065] 在通道子系统的情况下，子通道被用来执行 EADM 操作。这些子通道被称为异步数据移动器（ADM）类型的子通道，且与 EADM 操作有关，而不是 I/O 设备，像其他 I/O 类型的子通道那样。ADM 类型的子通道不包含设备号，也不包含通道路径信息。被提供给配置的 ADM 类型的子通道的号码是取决于模型的。ADM 类型的子通道由子系统识别字（SID）寻址。

[0066] ADM 类型的子通道被扩展异步数据移动器设施使用，该设施是通道子系统的扩展。如在此描述的，EADM 设施允许程序请求在主存储器和存储级内存之间的数据块的移动，以及执行其他操作，诸如清除存储级内存块或释放存储级内存块。在一个实施例中，当 EADM 设施被安装时：

- [0067] • 一个或多个 ADM 类型的子通道被提供,其被用于 EADM 操作。
- [0068] • EADM 操作被指定的 EADM 操作块(AOB) 指定。AOB 包括 EADM 请求块(ARQB) 和 EADM 响应块(ARSB),并指定 EADM 移动规格块(MSB) 的列表。对于移动操作,MSB 含有关于将被移动的数据块的信息,诸如块大小、块的源和目标位置以及数据移动的方向。
- [0069] • 可被 AOB 指定的 MSB 的最大数量是取决于模型的。MSB 可指定要被移动或操作的最大的块计数也是取决于模型的。
- [0070] • 程序通过发出指定 ADM 类型的子通道和 EADM 操作请求块(ORB) 的开始子通道指令而启动 EADM 操作。EADM ORB 转而指定 AOB。该指令将 EADM ORB 的内容传递给指定的子通道。
- [0071] • 当开始子通道被发出以启动 EADM 操作时,通道子系统异步地执行指定的操作。
- [0072] • 如由程序所观察到的,存储级内存在取决于模型的最小块大小上是块并行的。该取决于模型的值被称为 SCM 块并行大小。
- [0073] • 当 EADM 操作完成时,针对操作在其上被启动的 ADM 类型的子通道,使得 I/O 中断对程序待决。
- [0074] EADM ORB 包括子通道和将被使用的 AOB 的地址的指定。AOB 被指定为例如在 4 字节的边界上且大小可最多达 4K 字节。如果需要比适合于 4K 字节的 AOB 更多的 MSB,AOB 可指定一个 MSB 列表,其通过使用指定列表中的下一个 MSB 的地址、而不是指定将被用于数据传输的存储区域的 MSB,扩展到额外的存储区域。
- [0075] 使用指定的 EADM ORB 中的信息和指定的 AOB 中的 ARQB 以获得 MSB,由通道子系统开始第一个 EADM 操作。MSB 包括指定和控制将被处理的 EADM 操作的信息。
- [0076] 每个 EADM 操作被一个 MSB 表示。MSB 可指定例如从主存储器到存储级内存的数据块的传输;从存储级内存到主存储器的数据块的传输;存储级内存的块的清除;以及存储级内存的块的释放。
- [0077] 如果将被传送的存储块在主存储器中不是连续的,可使用新的 MSB,或者 MSB 可通过指定 EADM 间接数据地址字(AIDAW) 的列表以指定不连续的块,来使用间接寻址。
- [0078] 由于 MSB 仅指定一个方向的数据传输,当传输方向有变化时可使用新的 MSB。
- [0079] EADM 操作的结束通常是由通道端和设备端的组合状态条件指示的。这种状态组合表示主要和次要状态的组合,其指示在状态被清除后,子通道可用于另一个开始功能。
- [0080] EADM 操作可被清除子通道指令提前终止。清除子通道指令的执行终止于通道处的 AOB 的执行,从子通道清除执行中的 AOB 的指示,以及异步地执行清除功能。
- [0081] 以下将参考图 2A-2F 描述关于 EADM ORB 的进一步细节和相关控制结构。开始,参考图 2A,其描述了 EADM ORB 的一个实施例。
- [0082] 作为一个例子,EADM ORB200 包括:
- [0083] 中断参数 202:该字段在子通道中被保持为未修改,直到被随后的修改子通道或开始子通道指令替换。当针对子通道发生 I/O 中断时且当中断请求被例如测试待决中断的执行清除时,这些位被放置在 I/O 中断代码中。
- [0084] 子通道键 204:该字段形成了用于由 ARQB 指定的 EADM 操作的子通道键,并应用于获取 ARQB、获取 MSB、存储 ARSB 以及用于存取主存储器以便数据传输。该字段的值是定义的值;否则,程序检验条件被通道子系统识别或操作数异常被识别。

[0085] ORB 扩展控制(X) 205 :该字段指定 ORB 是否被扩展。当 ADM 类型的子通道被指定时,该字段是指定值;否则,识别操作数异常或程序检验条件。

[0086] EADM 操作块(AOB)地址 206 :该字段指定了 EADM 操作块(AOB)的地址。如果该字段的特定位不包括定义的值,则识别操作数异常或程序检验条件。

[0087] 如果该字段指定了被保护免于获取的位置或指定了配置外的位置,则不启动开始功能。在这种情况下,子通道变成以主要、次级和警告状态待决的状态。

[0088] 通道子系统(CSS)优先级 208 :该字段包括被分配给指定的子通道的通道子系统优先级号,并且当开始功能将针对一个或多个以待决开始的子通道而启动时被用来对 ADM 类型的子通道的选择进行排序。

[0089] 存储级内存(SCM)优先级 210 :该字段包括存储级内存优先级(SCM)号,其指定了被施加给与开始功能关联的所有 EADM 操作的优先级别。

[0090] SCM 优先级字段的内容是否被 EADM 设施识别是取决于模型的。在不识别该字段的模型上,字段内容被忽略,且与开始功能关联的所有 EADM 操作被分配一个隐含的优先级号码。

[0091] 格式(FMT)212: :该字段指定了 ORB 的布局。当 ADM 类型的子通道被指定时,该字段将包括指定值;否则,识别操作数异常或设置特定条件代码。

[0092] 由 EADM ORB 的 EADM AOB 地址 206 指定的 EADM 操作块(AOB)包括被用来发起 EADM 操作的信息。在一个例子中 AOB 被分配在 4K 字节的边界上且长度可变。

[0093] 在一个例子中,如图 2B 所示,EADM AOB220 包括三个部分:EADM 请求块(ARQB)222;EADM 响应块(ARSB) 224;以及包含高达定义数量(例如 124)的 MSB 的 MSB 区域 226。但是当使用 MSB 分支时(即,通过使用 MSB 中的分支到下一个 MSB 标志),ARQB 可指定使用多于定义的 MSB。

[0094] 参考图 2C 描述 EADM 请求块(ARQB) 222 的一个实施例。在一个例子中,ARQB222 包括:

[0095] 格式(FMT) 230 :该字段指定了 ARQB 的布局。该字段的值是定义的值;否则,程序检查条件被通道子系统识别,且命令代码错误在 ARSB 的异常限定器代码字段中被指示。

[0096] 命令代码 232 :该字段将指定 EADM 移动块命令;否则,程序检验条件被通道子系统识别,且命令代码错误在 ARSB 的异常修饰符代码字段中被指示。

[0097] MSB 计数 234 :该字段指定了组成 EADM 请求的 MSB 的计数。可被指定的最大的 MSB 计数是取决于模型的。该字段的值大于零且小于或等于取决于模型的最大 MSB 计数值;否则,程序检查条件被通道子系统识别,且 MSB 计数错误在 ARSB 的异常修饰符代码字段中被指示。

[0098] 除了 EADM 请求块,EADM ORB 也指定 EADM 响应块(ARSB)。在该实施例中,仅当异常条件被识别时,EADM 响应块是有意义的。特别地,仅当告警状态出现在 EADM 子通道状态字中(SCSW)、EADM 扩展状态字(ESW)是有意义的、且 EADM 响应块存储(R)位在 EADM 扩展报告字(ERW)中是 1(这些情况中的每个都将在以下描述)时,ARSB 是有意义的。当 ARSB 无意义时,AOB 中的 ARSB 的内容时不可预测的。

[0099] 如果当相关子通道是子通道活动的时程序存储到 ARSB,可能发生不可预测的结果。

- [0100] 当 ARSB 被存储时,已被传递的数据量(如有)是不可预测的。
- [0101] 参考图 2D 描述 EADM 响应块的一个实施例。在一个例子中,ARSB224 包括 :
- [0102] 格式(FMT) 240 :该字段指定了 ARSB 的布局。当 ARSB 被存储时,该字段的值被存储为定义的值。
- [0103] 异常标志(EF)242 :当 ARSB 被存储时,该字段,当被设置时,指明 ARSB 被存储的异常原因。示例性异常原因包括 :
- [0104] 程序检验 :检测到编程错误。
- [0105] 保护检验 :存储访问被保护机制阻止。保护适用于获取 ARQB、MSB、AIDAW 的获取,以及将被传送到存储级内存的数据,以及适用于在 ARSB 中存储信息以及从存储级内存传送的数据。
- [0106] 通道数据检验 :针对包含在主存储器中且当前被用于执行 EADM 操作的数据,已检测到不正确的存储错误。当检测到时可指示该条件,即使在预取时数据未被使用。当数据或相关键在数据被通道子系统引用时在主存储器中具有无效检验块代码(CBC)时,通道数据检验被指示。
- [0107] 通道控制检验 :通道控制检验是由影响通道子系统控制的任何机器故障引起的。该情况包括 ARQB、ARSB、MSB、AIDAW 或相应相关键上的无效 CBC。当在预取的 ARQB、MSB、AIDAW 或相应相关键上检测到无效 CBC 时,该条件可以被指示,即使该 ARQB、MSB 或 AIDAW 未被使用。
- [0108] 扩展异步数据移动设施检验 :关于包含在存储级内存中的、且当前在 EADM 操作的执行中被使用的数据已检测到不正确的错误。
- [0109] 异常控制块标识符(ECBI)244 :当存储 ARSB 时,该字段是多位掩码,其位当被设置时,指定与由 EF 字段指定的识别的异常关联的任何单个以下组件或其组合 :
- [0110] • 一个或多个控制块
 - [0111] • 主存储器数据区域
 - [0112] • 存储级内存
- [0113] 可被设置的位表示例如 EADM 移动规格块、EADM 间接数据地址、主存储器中的数据以及 / 或存储级内存中的数据。
- [0114] ECBI 字段中的位描述了与单个异常条件关联的组件。如果针对异常条件没有组件可被识别,该字段包含例如零。
- [0115] 字段有效性标志(FVF) 246 :当 ARSB 被存储时,该字段包括多位掩码,其位指示 ARSB 中特定字段的有效性。当有效性位被设置时,对应的字段已被存储且可用于恢复目的。可被设置的位表示例如失败的 MSB 地址字段、失败的 AIDAW 字段、失败的主存储器地址字段和 / 或失败的存储级内存地址字段。
- [0116] 异常修饰符代码(EQC) 248 :当 ARSB 被存储时,该字段包括代码值,其进一步描述了由异常标志字段指定的异常。代码值可表示以下,例如 :
- [0117] 不提供额外的描述。对于这种情况,异常控制块标识符(ECBI)字段和那些被字段有效性标志字段验证的字段可识别对其识别了异常的控制块。
- [0118] 格式错误 :由格式字段指定的格式被保存。对于这种情况,异常控制块标识符(ECBI)字段和那些被字段有效性标志字段验证的字段可识别已对其识别了异常的控制块。

- [0119] 命令代码错误 :在 ARQB 的命令代码字段中指定的值不被识别。
- [0120] MSB 计数错误 :在 ARQB 的 MSB 计数字段中指定的值是零或超过可被指定的取决于模型的 MSB 最大值。
- [0121] 标志错误 :由标志字段指定的标志位被保存。对于这种情况,异常控制块标识符(ECBI)字段和那些被字段有效性标志字段验证的字段识别已对其识别了异常的控制块。
- [0122] 操作代码错误 :指定保存的操作代码值。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段含有已对其识别了异常的 MSB 的地址。
- [0123] 块大小错误 :指定保存的块大小值。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。
- [0124] 块计数错误 :在 MSB 的块计数字段中指定的值是零或超过可被 MSB 指定的取决于模型的块的最大计数。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。
- [0125] 主存储器地址规格错误 :在不正确的边界上指定了主存储器地址。这样的地址可能已被 MSB 指定或被 AIDAW 指定。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的主存储器地址字段是有效的,字段包含已对其识别了异常的主存储器地址。
- [0126] 存储级内存地址规格错误 :在不正确的边界上指定了存储级内存地址。这样的地址被 MSB 指定。如果字段有效性标志指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的存储级内存地址字段是有效的,该字段包含已对其识别了异常的存储级内存地址。
- [0127] 主存储器地址异常 :EADM 设施尝试使用在配置中不可用地址或回绕(wrap)最大存储器地址。这样的地址可能已被 MSB 指定,或来自于数据传送期间增加主存储器地址。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 地址。如果字段有效性标志字段指示失败的主存储器地址字段是有效的,该字段包含已对其识别了异常的主存储器地址。
- [0128] 存储级内存地址异常 :EADM 设施尝试使用在配置中不可用的存储级内存地址。这样的地址可能已被 MSB 指定,或产生于在数据传送期间增加存储级内存地址。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的存储级内存地址字段是有效的,该字段包含已对其识别了异常的存储级内存地址。
- [0129] 主存储器错误 :检测到不正确的主存储器错误。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志指示失败的主存储器地址字段是有效的,该字段包含已对其

识别了异常的主存储器地址。

[0130] MSB 列表错误 :AOB 指定跨过 4K 字节边界的 MSB 列表,而不指定跨过边界的分支到下一个 MSB (BNM),或者 MSB 是指定的 MSB 列表中的最后一个 MSB,且 BNM 由该 MSB 指定。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。

[0131] MSB 分支错误 :AOB 指定 MSB 列表,其中 MSB (分支源 MSB) 指定分支到下一个 MSB (BNM),以及作为分支目标的 MSB 也指定 BNM。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的分支源 MSB 的地址。

[0132] AIDAW 列表错误 :MSB 指定跨过 4K 字节边界的 EADM 间接数据地址字(AIDAW)列表,而没有指定跨过边界的分支到下一个 AWIDAW (BNA)。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。

[0133] AIDAW 分支错误 :MSB 指定 EADM 间接数据地址字(AIDAW)列表,其中 AIDAW (分支源 AIDAW)指定了分支到源 AIDAW (BNA)且作为分支目标的 AIDAW 也指定了 BNA。如果字段有效性标志字段指定失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的分支源 AIDAW 的地址。

[0134] 存储级内存临时错误 :检测到可恢复的存储级内存错误。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址时有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的存储级内存地址字段是有效的,该字段包含已对其识别了异常的存储级内存地址。

[0135] 释放操作不支持错误 :针对释放操作不被支持的存储级内存,指定了释放操作。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的存储级内存地址字段是有效的,该字段包含已对其识别了异常的存储级内存地址。

[0136] 释放数据读取错误 :针对最后成功的操作是释放操作的存储级内存,指定了读取操作。如果字段有效性标志字段指示失败的 MSB 地址字段是有效的,该字段包含已对其识别了异常的 MSB 的地址。如果字段有效性标志字段指示失败的 AIDAW 地址字段是有效的,该字段包含已对其识别了异常的 AIDAW 的地址。如果字段有效性标志字段指示失败的存储级内存地址字段是有效的,该字段包含已对其识别了异常的存储级内存地址。

[0137] 失败的 MSB 地址 250 :当设置了字段有效性标志字段的指定位时,该字段指示已对其识别了异常的 MSB 的地址。

[0138] 失败的 AIDAW 地址 252 :当设置了字段有效性标志字段的指定位时,该字段指示已对其识别了异常的 AIDAW 的地址。

[0139] 失败的主存储器地址 254 :当设置了字段有效性标志字段的指定位时,该字段指示已对其识别了异常的主存储器的块的地址。

[0140] 失败的存储级内存地址 256 :当设置了字段有效性标志字段的指定位时,该字段包括已对其识别了异常的存储级内存的块的(例如 64 位的) SMC 地址。

[0141] 除了 EADM 请求块和 EADMD 响应块, EADM 操作块(AOB)也指定一个或多个 EADM 移动规格块(MSB)。AOB 可包含多达例如 124 个 MSB。但是,程序可通过在 ARQB 的计数字段中指定更大数量的 MSB 并通过使用分支到下一个 MSB (BNM) 标志来分支到 MSB 列表的延续,来指定超过 124 个的 MSB。可能存在多个 MSB 列表的延续,但是在任何一个实施例中,这些延续中没有一个可跨过 4K 字节的边界而不使用 BNM 来跨过该边界。MSB 的延续不要求是连续的。MSB 列表的每个延续,如果被指定为在 4K 字节的边界上开始,可包含多达 128 个 MSB。MSB 列表中的 MSB 总数由 ARQB 中的 MSB 计数字段指定。

[0142] EADM 移动规格块描述例如将在主存储器和存储级内存之间被移动的存储器块,或将在存储器块上被执行的操作。MSB226 的一个实施例参考图 2E 被描述,且包括例如 :

[0143] 格式(FMT) 260 :该字段指定 MSB 的布局。该字段的值是定义的值;否则,在 ARSB 的异常修饰符代码字段中指示 MSB 格式错误。

[0144] 操作代码(OC)262 :该字段指定将被执行的操作。在一个例子中,当如下所述标志字段的指定位是定义的值时,该字段指定将被执行的操作。示例性操作包括 :

[0145] 清除存储级内存 :当该代码被指定时,由存储级内存地址、块大小和块计数字段指定的存储级内存被清除(例如被设置为零)。主存储器地址字段对于该操作无意义。

[0146] 读取 :当该代码被指定时,数据被指定为从存储级内存传递到主存储器。块计数、块大小、主存储器地址和存储级内存地址字段对于该操作有意义。

[0147] 写入 :当该代码被指定时,数据被指定为从主存储器传递到存储级内存。块计数、块大小、主存储器地址和存储级内存地址字段对该操作有意义。

[0148] 释放 :当该代码被指定且 EADM 释放设施被安装时,由存储级内存地址、块大小和块计数字段指定的存储级内存被释放。主存储器地址字段对该操作无意义。当 EADM 释放设施未被安装时,该代码被保留。

[0149] 如果保留值被指定,程序检验条件被通道子系统识别,且操作代码错误在 ARSB 的异常修饰符代码字段中被指示。

[0150] 当标志字段的指定标志位不是定义的值时,该字段被保留且不被检验。

[0151] 标志 264 :该字段识别可被设置的一个或多个标志。被保留的标志位被设为定义的值;否则,标志错误在 ARSB 的异常修饰符代码字段中被指示。示例性标志包括 :

[0152] • 分支到下一个 MSB (BNM) :当被设置时,该标志指示 MSB 不指定 EADM 操作,且不被用来传送数据。反而,主存储器地址字段指定将被使用以指定 EADM 操作的下一个 MSB 的地址。

[0153] 当 BNM 标志被设置(例如为 1)且主存储器地址字段指定其中也被设置了 BNM 标志的 MSB 时,程序检验条件被通道子系统识别,分支源 MSB 的地址被存储在 ARSB 的失败的 MSB 地址字段中,且 MSB 分支错误在 ARSB 的异常修饰符代码字段中被指示。

[0154] 如果 MSB 在 4K 字节的边界结束, AOB 指定额外的 MSB, 且 BNM 标志未被设置, 程序检验条件被通道子系统识别, MSB 的地址被存储在 ARSB 的失败的 MSB 地址字段中, 且 MSB 列表错误在 ARSB 的异常修饰符代码字段中被指示。

[0155] 如果 MSB 是指定的 MSB 列表中的最后一个 MSB 且 BNM 标志被设置, 程序检验条件

被通道子系统识别, MSB 的地址被存储在 ARSB 的失败的 MSB 地址字段中, 且 MSB 列表错误在 ARSB 的异常修饰符代码字段中被指示。

[0156] 当 BNM 标志被设置时, 剩余的标志、操作代码字段、块大小字段、块设置计数字段和存储级内存地址无意义。

[0157] • 间接寻址 :当被设置时, 该标志指示主存储器地址字段指定 EAM 间接数据地址字 (AIDAW) 的地址或指定分别将被用于数据传送的一个或多个主存储器位置的 AIDAW 列表的第一个 AIDAW 的地址。当未设置时, 该标志指示主存储器地址字段指定将被用于数据传送的主存储器位置的地址。

[0158] 块大小 (BS) 266 :该字段指定将针对主存储器和存储级内存两者被传送的数据块的大小和边界, 或将被操作(例如清除或释放)的存储级内存的块的大小和边界。

[0159] 4K :当代码值是一个定义的值时, 将被传送的数据块是在 4K 字节边界上且大小是 4K 字节。

[0160] 1M :当代码值是另一个定义的值时, 将被传送的数据块是在 1M 字节边界上且大小是 1M 字节。

[0161] 如果指定了保留值, 程序检验条件被通道子系统识别, 且块大小错误在 ARSB 的异常修饰符代码字段中被指示。

[0162] 当 BNM 标志被设置时, 该字段被忽略。

[0163] 块计数 268 :该字段包括将被传送或被操作(例如清除或释放)的数据块的计数。基于块大小字段, 这是 4K 字节或 1M 字节块的计数。

[0164] 该字段的值将大于零, 并小于或等于可由 MSB 指定的取决于模型的最大块计数; 否则, 程序检验条件被通道子系统识别, 且块计数错误在 ARSB 的异常修饰符代码字段中被指示。

[0165] 当 BNM 标志被设置时, 该字段被保留且不被检验。

[0166] 主存储器地址 270 :当操作代码字段有意义(例如 BNM 标志未被设置)并指定读或写操作, 且标志字段中的间接寻址指示符未被设置时, 该字段包括将被用于数据传送的主存储器地址, 且以下各项适用 :

[0167] • 当块大小字段指定 4K 字节的块时, 该字段的指定位包含定义的值(例如零); 否则, 程序检验条件被通道子系统识别, 且主存储器规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0168] • 当块大小字段指定 1M 字节块时, 该字段的指定位包含定义的值(例如零); 否则, 程序检验条件被通道子系统识别, 且主存储器规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0169] 当操作代码字段有意义并指定读或写操作, 且标志字段中的间接寻址指示符被设置时, 该字段包括将被用于数据传送的 AIDAW 的主存储器地址或 AIDAW 列表中的第一个的地址。对于这种情况, 该字段的特定位包含定义的值; 否则, 程序检验条件被通道子系统识别, 且主存储器规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0170] 当 BNM 标志被设置时, 操作代码无意义, 且 MSB 不指定 EADM 操作。反而, 该字段包括指定 EADM 操作的下一个 MSB 的主存储器地址。对于这种情况, 该字段的指定位包含定义的值; 否则, 程序检验条件被通道子系统识别, 且主存储器规格错误在 ARSB 的异常修饰符

代码字段中被指示。

[0171] 存储级内存地址 272 :当操作代码字段有意义时,该字段包括将被用于数据传送或将被操作(例如清除、释放)的(例如 64 位)存储级内存地址,且以下各项适用:

[0172] • 当块大小字段指定 4K 字节的块时,该字段的指定位包含定义的值(例如零);否则,程序检验条件被通道子系统识别,且存储级内存规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0173] • 当块大小字段指定 1M 字节块时,该字段的指定位包含定义的值;否则,程序检验条件被通道子系统识别,且存储级内存规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0174] 当 BNM 标志被设置时,该字段被保留且不被检验。

[0175] 如上指示,EADM 间接数据地址字可被指定。EADM 间接数据地址字(AIDAW)允许程序指定在存储级内存和主存储器的非连续块之间的数据块传送。当 MSB 中的间接寻址标志被设置时,一个 AIDAW 或 AIDAW 列表由 MSB 指定。

[0176] 由单个 AIDAW 传送的数据量由 MSB 中的块大小字段指定。AIDAW 列表中 AIDAW 的数量是 MSB 中的块计数字段指定的数量加上指定分支到下一个 AIDAW 的 AIDAW 的数量的和。数据传送可以与在 AIDAW 列表中指定的顺序不同的顺序被处理。而且,由 AIDAW 列表中的多个 AIDAW 指定的数据传送可被并行处理。

[0177] AIDAW 在例如四字边界上被分配。AIDAW 列表可以是任何长度,但在一个例子中不跨过 4K 字节边界,除非指定了分支到下一个 AIDAW(BNA)来跨过边界。不要求作为分支目标的 AIDAW 与指定 BNA 的 AIDAW 是连续。但是,程序应当在尽可能少的 4K 字节块中创建 AIDAW 列表;否则,可能发生性能下降。

[0178] 参考图 2F,在一个例子中,EADM 间接数据地址字 280 包括:

[0179] 标志 282 :示例性标志包括:

[0180] 分支到下一个 AIDAW(BNA):当被设置时,该标志指示主存储器地址字段不指定将被用于传送数据的主存储器地址。反而,主存储器地址字段指定将被用来传送数据的下一个 AIDAW 的地址。

[0181] 当 BNA 标志被设置且主存储器地址字段指定了其中 BNA 标志也被设置的 AIDAW 时,MSB 的地址被存储在 ARSB 的失败的 MSB 地址字段中,分支源 AIDAW 的地址被存储在 ARSB 的失败的主存储器地址字段中,且 AIDAW 分支错误在 ARSB 的异常修饰符代码字段中被指示。

[0182] 如果 AIDAW 在 4K 字节边界处结束,MSB 指定额外的 AIDAW,且 BNA 标志未被设置,则 MSB 的地址被存储在 ARSB 的失败的 MSB 地址字段中,AIDAW 的地址被存储在 ARSB 的失败的主存储器地址字段中,且 AIDAW 列表错误在 ARSB 的异常修饰符代码字段中被指示。

[0183] 主存储器地址 284 :当 BNA 字段未被设置时,该字段包括将被用于数据传送的主存储器地址,且以下各项适用:

[0184] • 当 MSB 中的块大小字段指定 4K 字节的块时,该字段的指定位包含定义的值(例如零);否则,程序检验条件被通道子程序识别,且主存储器规格错误在 ARSB 的异常修饰符代码字段中被指示。

[0185] • 当 MSB 中的块大小指定 1M 字节的块时,该字段的指定位包含定义的值(例如零);否则,程序检验条件被通道子程序识别,且主存储器规格错误在 ARSB 的异常修饰符代码字

段中被指示。

[0186] •当BNA标志被设置时,该字段包括将被用于数据传送的下一个AIDAW的主存储器地址。

[0187] 如上所述, EADM 操作被开始子通道指令指定。即,程序通过发出指定 ADM 类型的子通道和 EADM ORB 的开始子通道指令而启动 EADM 操作。指令的执行将 EADM ORB 的内容传递给指定的子通道。EADM ORB 包括子通道键的指定(用于保护)和将被使用的 AOB 的地址。

[0188] 在一个例子中,如图 3A 所示,开始子通道指令 300 包括指定开始子通道功能的操作代码 302、第一操作数 304,其是位于例如通用寄存器 1 中的暗示的操作数,其包括指定将被开始的 ADM 类型的子通道的子系统标识符;以及第二操作数 306,其是 EADM ORB 的逻辑地址。EADM ORB 指定用于控制开始功能的参数。在开始子通道的执行期间,在设置条件代码前,EADM ORB 的内容被放置在指定的子通道中。如果开始子通道的执行导致设置了指示成功的代码之外的条件代码,EADM ORB 的内容不被放置在指定的子通道中。

[0189] 在用于 ADM 类型的子通道的开始子通道的执行后,通道子系统异步地执行开始功能以用 EADM 设施启动 EADM 操作。开始功能包括例如以下各项:

[0190] 1. 从 AOB 获取 ARQB。

[0191] 2. 执行由 ARQB 及其指定的 MSB 指定的 EADM 操作。

[0192] 3. 有条件地将 ARSB 中的完成信息存储在 AOB 中。

[0193] 4. 使得 ADM 类型的子通道变成状态待决,指示开始功能的完成。

[0194] 在一个实施例中,参考图 3B,当开始子通道执行令被执行且 SID 指定 ADM 类型的子通道且第二操作数指定 EADM ORB,在步骤 320 指定 EADM 操作。在步骤 322 EADM ORB 中的参数被传递给指定的子通道,且在步骤 324 通道子系统被请求以利用 EADM 设施执行开始功能。在步骤 326,通道子系统通过使用子通道处的信息(包括在开始子通道指令的执行期间传递的信息)启动 EADM 操作,来异步地执行开始功能。

[0195] 在步骤 328, EADM 操作(例如第一操作)的执行包括使用来自 EADMORB 的信息获得 AOB,且在步骤 330 该信息被用在 AOB 中以获得 EADM 请求块(ARQB)以及指定一个或多个 EADM 移动规格块(MSB)。然后在步骤 332 一个或多个指定的 MSB 从主存储器被获取,并且在 MSB 中指定的信息被用于控制所请求的 EADM 操作。当通道子系统试图开始数据传送或试图清除或释放操作时,第一操作被认为开始了。

[0196] 在步骤 334 通道子系统执行 MSB 中请求的操作。该处理与开始处理的执行是异步的。

[0197] 当在步骤 336 被开始子通道启动的 EADM 操作结束时,通道子系统在步骤 338 产生状态条件。在步骤 339,通过 I/O 中断使这些条件的生成引起程序的注意。程序也会通过执行测试待决中断指令来查询这些条件。

[0198] 生成的状态条件以 EADM 子通道状态字(SCSW)的形式被呈现给程序。通过执行测试子通道指令,EADM SCSW 被存储为 EADM 中断响应块(IRB)的一部分。

[0199] 当在 EADM 操作块(AOB)中指定的 EADM 操作被结束时,通道子系统产生初级和次级中断状态。EADM 操作可被清除子通道终止或者在执行开始功能时异常条件被识别。

[0200] 对于移动操作,由通道子系统进行的 MSB 的处理控制信息单位到主存储器或来自

主存储器的流。为了改变 MSB 处理期间数据移动的方向,需要新的 MSB。ARQB 指定构成请求的 MSB 的计数。

[0201] 每个 EADM 操作由一个 MSB 表示。MSB 可指定以下各项中的任一个：

[0202] • 从主存储器到存储级内存的数据块的传送。

[0203] • 从存储级内存到主存储器的数据块的传送。

[0204] • 清除存储级内存的块。

[0205] • 当安装了 EADM 释放设施时,释放存储级内存的块。

[0206] 如果将为移动操作而传送的存储块在主存储器中不是连续的,可使用新的 MSB,或者 MSB 可通过指定 EADM 间接数据地址字(AIDAW)列表而使用间接地址,以指定不连续块。由于 MSB 可仅在一个方向指定数据传送,当数据传送方向有变化时要使用新的 MSB。

[0207] 以下是 EADM 数据传送的特点：

[0208] • 数据传送可不按指定的 MSB 列表的顺序被处理。

[0209] • 由 MSB 列表中的多个 MSB 指定的数据传送可被并行处理。

[0210] • 数据传送可以不按指定 AIDAW 列表的顺序被处理。

[0211] • 在 AIDAW 列表中的多个 AIDAW 中指定的数据传送可被同时处理。

[0212] • 对主存储器和存储级内存的访问不必是单一访问引用,且不必必须以从左到右的方向被执行,如程序和其他 CPU 所观察到的。

[0213] • 如果两个或更多个 EADM 操作当前是活动的并寻址相同的 SCM 位置、主存储器位置或这两者,操作可被同时执行且来自不同操作的内容可被交错;但是:

[0214] 一对于输入操作,被 EADM 设施存储到主存储器的每个块的、其大小等于 SCM 块并行大小的数据由仅通过一个并行 EADM 操作而从存储级内存传送的数据组成。

[0215] 一对于输出操作,存储级内存的位于边界上的、且大小等于 SCM 块并行大小的每个块包含仅被一个并行 EADM 操作指定的数据。

[0216] 以上是真实的,而不考虑 EADM 操作是否被单个 AOB 指定,且被开始功能的相同实例处理,或 EADM 操作被不同的 AOB 指定,且被开始功能的不同实例处理。

[0217] • 当 EADM 操作对于子通道是活动的时,无法预测是否由程序对 ARQB、MSB、AIDAW 和与该活动的操作关联的传送数据作出的更改被 EADM 设施观察到。

[0218] 当被 AOB 指定的所有 MSB 指定的所有块已被传送或清除或释放时,子通道生成状态,其被存储在子通道中,并请求用于 ADM 类型的子通道的 I/O 中断。

[0219] EADM 操作的结束通常由通道和设备终端的组合状态条件指示。该状态组合表示初级和次级状态的组合,其指示状态被清除后,子通道可用于另一个开始功能。

[0220] 如上所述,MSB 可指定数据传送操作、清除操作或释放操作,每种操作都在以下被描述。

[0221] 当请求移动操作时,一个或多个数据块在主存储器和存储级内存之间移动。例如,对于读取操作,SCM 数据从提供由指定的 SCM 地址指定的内容的 SSD 获取,随后该内容被存储在主存储器中。对于写操作,该过程被保留。控制 SSD 的适配器执行存储。在进一步的细节中,为了执行移动操作,系统固件首先将给定 SCM 地址转换为适配器地址(例如逻辑卷地址,其中逻辑卷包括一个或多个 SSD)。例如,将 SCM 地址与适配器地址关联的转换表被使用。系统固件随后向一个或多个 I/O 适配器提交一个或多个对应的适配器移动命令(例

如读或写)。适配器移动命令包含主存储器地址、适配器地址和传送大小。适配器随后使用另一个转换表来定位一个或多个物理 SSD 地址,其对应于适配器地址。I/O 适配器通过从主存储器获取数据并将其存储在 SSD 中,或从 SSD 获取数据并将其存储在主存储器中来实现移动操作。在文档号为 POU920110090US1 的、标题为“Firmware Management of Storage Class Memory”(存储级内存的固件管理)的共同提交的专利申请中进一步提供额外的细节,该文献在此通过引用全文并入本文。

[0222] 当执行清除操作时,通过将内容设为零,指定的存储级内存增量被清除。

[0223] 而且,当释放操作被执行时,一个或多个标准 TRIM 命令可被提交给包含对应的 SCM 地址的 SSD。TRIM 命令允许程序给出关于块使用的暗示,从而允许由 SSD 进行的更好的垃圾收集。TRIM 命令允许操作系统通知 SSD,哪些存储块不再被认为在使用中且可被消除。

[0224] EADM 操作可被清除子通道指令提前中断。清除子通道指令的执行终止子通道处的 AOB 的执行,从子通道清除执行中的 AOB 的指示,并异步地执行清除功能。当清除功能被执行时,在子通道变成状态待决之前,数据传送被终止且传送的数据量是不可预测的。清除功能的执行不会导致状态生成,但是会引起通道子系统使得 I/O 中断待决。

[0225] 在一个实施例中,参考图 3C,清除子通道指令 350 包括操作代码 352,其指定清除子通道功能。将被清除的子通道被例如通用寄存器中的子系统识别字指定。用于 ADM 类型的子通道的清除功能包括:

[0226] 1. 保证当前数据传送被终止。

[0227] 2. 修改子通道处的字段以及有条件地修改 ARSB。例如,子通道状态字被修改为在功能控制字段中和活动的控制字段中指示清除功能。ARSB 可被修改为反应任何检测到的错误。

[0228] 3. 使得子通道变为状态待决,指示清除功能的完成。

[0229] 也可发出可指定 ADM 类型的子通道的其他指令,包括测试子通道、修改子通道和存储子通道,每种指令都在以下描述。

[0230] 参考图 3D,在一个例子中,测试子通道指令 360 包括例如操作代码 362,其指定测试子通道功能;第一操作数 364,其是位于例如通用寄存器 1 中的暗示的操作数,其包含指定将被测试的子通道的子系统识别字;以及第二操作数 366,其是其中存储了信息的信息响应块(IRB)的逻辑地址。

[0231] 当测试子通道被执行,且指定 ADM 类型的子通道,子通道是状态待决的,且信息被存储在指定的 EADM IRB (中断响应块) 中时,设置指定的条件代码。当子通道不是状态待决的且状态信息被存储在指定的 EADMIRB 中时,定义的条件代码被设置。当子通道未被提供或未被使能时,不采取动作。

[0232] 在一个例子中,EADM IRB 包括子通道状态字(SSW) 和扩展状态字(ESW),以及可提供额外的取决于模型的信息的扩展控制字,其描述可能在设施中存在的条件。在讨论可指定 ADM 类型的子通道的各种指令后,这些字的每个在以下被进一步描述。

[0233] 参考图 3E,在一个实施例中,修改子通道指令 370 包括操作代码 372,其指示修改子通道功能;第一操作数 374,其是位于例如通用寄存器 1 中的暗示的操作数,其包括指定将被修改的子通道的子系统识别字;以及第二操作数 376,其是与子通道关联的子通道信息块(SCHIB)的逻辑地址。

[0234] 当修改子通道被执行且指定 ADM 类型的子通道,且来自指定的子通道信息块(SCHIB)的信息被放置在子通道中时,设置特定条件代码。当子通道是状态待决的,不采取动作,且设置定义的状态代码。当子通道忙于开始或清除功能时,不采取动作。

[0235] 参考图 3F,描述了存储子通道指令的一个例子。在一个例子中,存储子通道指令 380 包括操作代码 380,其识别存储子通道功能;第一操作数 384,其是位于例如通用寄存器 1 中的暗示的操作数,其包括指定信息正被存储的子通道的子系统识别字;以及第二操作数 386,其是 SCHIB 的逻辑地址。

[0236] 当存储子通道被发出且指定 ADM 类型的子通道,且 SCHIB 被存储时,指定的条件代码被设置。当指定的子通道在通道子系统中未被提供时,不采取动作。

[0237] 参考图 4A 描述了用于 ADM 类型的子通道 EADM (SCHIB)的子通道信息块的一个例子。在一个例子中,EADM SCHIB400 包括取决于模型的区域 401,其包括取决于模型的信息。而且,SCHIB400 包括路径管理控制字(PMCW) 402,以及子通道状态字(SCSW) 404,其每种将在以下被描述。

[0238] 在一个实施例中,EADM PMCW402 包括例如在图 4B 中示出的以下字段:

[0239] 中断参数 410 :该字段包括被存储在 I/O 中断代码中的中断参数。中断参数可被设开始子通道和修改子通道设为任何值。子通道中的中断参数字段的初始值是零。

[0240] 中断子类(ISC) 412 :该字段包括多个位,其是指定范围内的无符号的二进制整数,对应于配置中的每个 CPU 的指定控制寄存器中的 I/O 中断子类掩码位的位位置。在 CPU 的控制寄存器中的掩码位的设置控制了该 CPU 对与子通道关联的中断请求的识别。ISC 可被修改子通道设为一个值。子通道中的 ISC 字段的初始值是例如零。

[0241] 使能(E) 414 :当被设置时,该字段指示子通道被使能用于所有 EADM 功能。

[0242] 子通道类型(ST)416 :该字段指示子通道类型。根据模型和配置,一个或多个以下子通道类型可被提供:I/O 子通道或 ADM 子通道。

[0243] 当子通道被配置时该字段的值被确定,且不能被修改子通道改变。

[0244] 当修改子通道指令被执行并制定 ADM 类型的子通道时,ST 将指示 ADM 子通道;否则,操作数异常被识别。

[0245] 参考图 4A,子通道信息块也包括子通道状态字 404。EADM 子通道状态字(SCSW)提供了对程序的指示,其描述了 ADM 类型的子通道和相关 EADM 操作的状态。在一个例子中,如图 4C 所示,子通道状态字 404 包括:

[0246] 子通道键(键)420 :当功能控制字段(如下描述的)中的 EADM 开始功能指示符被设置时,该字段包括被通道子系统使用的存储访问键。当开始子通道被执行时,这些位与在 EADM ORB 中指定的键相同。

[0247] 扩展状态字格式(L)422 :当状态控制字段(如下描述的)的状态待决指示符被设置时,该字段,当被设置时,指示格式 -0ESW 已被存储。当包含以下任何指示的中断条件被测试子通道清除时,格式 -0ESW 被存储:

[0248] • 程序检验

[0249] • 保护检验

[0250] • 通道数据检验

[0251] • 通道控制检验

[0252] • 扩展的异步数据移动设施(EADMF)

[0253] 延迟条件代码(CC)424:当 EADM 开始功能指示符被设置且状态待决指示符也被设置时,该字段指示当测试子通道或存储子通道被执行时子通道是状态待决的一般原因。当子通道以任何状态组合是状态待决的,且仅当 SCSW 中的功能控制字段的开始功能指示符被设置时,延迟条件代码是有意义的。

[0254] 如被设置,延迟条件代码被用来指示是否遇到在子通道是开始待决时阻止子通道变为子通道活动的情况。

[0255] 示例性延迟条件代码包括:

[0256] • 正常的 I/O 中断已被呈现。

[0257] • 状态存在于 EADM SCSW 中,其是由通道子系统为阻止 EADM 开始功能被成功启动的情况生成的。即,子通道不会转换到子通道活动状态。

[0258] CCW 格式(F)426:当 EADM 开始功能指示符被设置时,该字段被存储为定义的值。

[0259] 预获取(P)428:当 EADM 开始功能指示符被设置时,该字段被存储为定义的值。

[0260] 扩展控制(E)430:当被设置时,该字段指示取决于模型的信息被存储在 EADM 扩展控制字(ECW)中。

[0261] 功能控制(FC)432:功能控制字段指示在子通道处被指示的 EADM 功能。示例性功能包括:

[0262] • EADM 开始功能:当被设置时,指示 EADM 开始功能已被请求,且在 ADM 类型的子通道处于待决或进行中状态。当对于开始子通道成功的条件代码被设置时,EADM 开始功能在子通道中被指示。当测试子通道被执行且子通道是状态待决的时,EADM 开始功能在子通道被清除。在清除子通道的执行期间,EADM 开始功能也在子通道处被清除。

[0263] • EADM 清除功能:当被设置时,指示 EADM 清除功能已被请求且在 ADM 类型的子通道处于待决或进行中状态。当为清除子通道设置了成功的条件代码时,EADM 清除功能在子通道处被指示。当测试子通道被执行且子通道是状态待决的时,EADM 清除功能指示在子通道处被清除。

[0264] 活动控制(AC)434:活动控制字段指示之前在子通道处接受的 EADM 功能的当前进度。

[0265] 当测试子通道被执行且子通道是状态待决的时,在活动控制字段中由位表示的所有条件在 ADM 类型的子通道处被重置。

[0266] 示例性活动包括:

[0267] • 开始待决:当被设置时,指示子通道是开始待决的。通道子系统可能处于或不处于执行 EADM 开始功能的过程中。当为开始子通道设置成功的条件代码时,子系统变为开始待决的。当执行 EADM 开始功能且通道子系统确定阻止获取 ARQB 的条件存在时,子通道保持开始待决。

[0268] 当以下各项中的任何一个发生时,子通道不再是开始待决的,

[0269] • 通道子系统试图启动由 AOB 指定的第一个数据传送。

[0270] • 清除子通道被执行。

[0271] • 测试子通道清除子通道处的状态条件。

[0272] 清除待决:当被设置时,子通道是清除待决的。通道子系统可以处于或不处于执行

EADM 清除功能的过程中。当为清除子通道设置指定的条件代码时,子通道变成清除待决的。

[0273] 当以下各项中的任一个发生时,子通道不再是清除待决的:

[0274] • EADM 清除功能被执行。

[0275] • 测试子通道单独清除状态待决条件。

[0276] 子通道活动的:当被设置时,指示 ADM 类型的子通道是子通道活动的。当通道子系统试图启动由 AOB 指定的第一个数据传送或执行第一个操作(看哪个先发生)时,ADM 类型的子通道被称为是子通道活动的。

[0277] 当以下各项中的任何一个发生时,子通道不再是子通道活动的:

[0278] • 子通道变成状态待决。

[0279] • 执行了清除子通道。

[0280] 状态控制(SC) 436:状态控制字段向程序提供了由子通道状态和设备状态字段中的信息描述的中断条件的摘要级指示。多于一个的状态控制指示符可被设置,作为子通道处的条件的结果。

[0281] 示例性状态控制包括:

[0282] 告警状态:当被设置时,指示存在告警中断条件。当告警状态出现在子通道时,告警中断条件被识别。在以下任何条件下告警状态被通道子系统生成:

[0283] • ADM 类型的子通道是开始待决的,且状态条件阻止第一个数据传送的启动。

[0284] • 子通道是子通道活动的,且被报告为子通道状态的异常条件已终止了 EADM 操作。

[0285] 当测试子通道或清除子通道被执行时,告警状态在子通道处被清除。

[0286] 初级状态:当被设置时,指示存在初级状态。当初级状态出现在子通道时,初级中断条件被识别。初级中断条件是请求的中断条件,当伴随有次级中断条件时,其指示 EADM 开始功能在子通道处完成。

[0287] 当执行测试子通道或清除子通道时,初级中断条件在子通道处被清除。

[0288] 次级状态:当被设置时,指示存在次级中断条件。当次级状态出现在子通道时,次级中断条件被识别。次级中断条件时请求的中断条件,当伴随有初级中断条件时,其指示子通道处的 EADM 开始功能的完成。

[0289] 当测试子通道或清除子通道被执行时,次级中断条件在子通道处被清除。

[0290] 状态待决:当被设置时,指示子通道是状态待决的,且描述中断条件的原因的信息是可用的。当测试子通道被执行从而以设置的状态待决位存储 EADM SCSW 时,所有的 EADM SCSW 指示在子通道处被清除,从而将子通道置于空闲状态。在清除子通道的执行期间,状态待决条件也在子通道处被清除。

[0291] 当状态待决被设置时,所有对用于子通道 ARSB 的主存储器和存储级内存的访问都被终止。

[0292] 子通道状态 438:ADM 类型的子通道状态条件被通道子系统检测并存储在子通道状态字段中。当子通道是状态待决的时,子通道状态字段是有意义的。除了由设备故障引起的条件,仅当通道子系统涉及 EADB 功能的处理时,可产生子通道状态。

[0293] 状态条件的例子包括:

[0294] • 程序检验:当编程错误被通道子系统检测到时,发生程序检验。

[0295] • 保护检验 :当通道子系统试图进行被保护机制阻止的存储访问时,保护检验发生。保护适用于获取 ARQB、MSB、AIDAW,以及将被传送给存储级内存的数据,以及适用于 ARSB 中的信息存储和从存储级内存传送的数据。

[0296] • 通道数据检验 :通道数据检验指示针对从主存储器的数据获取或将数据存储到主存储器中而检验到不正确的存储错误。

[0297] • 通道控制检验 :通道控制检验指示针对 AOB、MSB 或 AIDAW 的获取或存储检测到不正确的存储错误,或通道子系统遇到机器故障且故障影响 EADM 操作。

[0298] • 扩展的异步数据移动(EADM)设施检验 :EADM 设施检验指示针对到存储级内存的数据传送或来自存储级内存的数据传送,或针对在存储级内存上执行操作,错误被 EADM 设施检验到。

[0299] 当程序检验、保护检验、通道数据检验、通道控制检验或 EADM 设施检验条件被通道子系统识别时,EADM 操作被终止,且通道变成状态待决的,具有初级、次级和告警状态。

[0300] EADM 操作块地址 440 :该字段指示 EADM 操作块地址。

[0301] 设备状态 442 :这包括设备端或通道端。

[0302] 当子通道状态字的 ESW 格式 422 被设置,且子通道是状态待决的,提供 EADM 子通道扩展状态字(EADM ESW),其指定关于 ADM 类型的子通道的额外信息。

[0303] 在一个例子中,参考图 4D, EADM 子通道扩展状态字 450 包括 :

[0304] EADM 子通道注销 452 :在一个例子中, EADM 子通道注销包括 :

[0305] • 扩展状态标志(ESF) :当其位是一时,该字段指定错误已被通道子系统检测到。

[0306] 扩展状态标志的例子包括 :

[0307] 键检验 :当被设置时,指示通道子系统已检测到当引用 EADM 操作块(AOB)、EADM 移动规格块(MSB)、或 EADM 间接数据地址字中(AIDAW)中的数据时相关存储键上的无效的检验块代码(CBC)。

[0308] • AOB 地址有效性 :当被设置时,指示存储在 SCSW 的 AOB 地址字段中的地址不能被用于恢复目的。

[0309] EADM 扩展报告字 454,其包括例如 :

[0310] EADM 操作块错误指示符,当被设置时,指示存储在 EADM SCSW 中的异常状态与指定的 EADM 操作块(AOB)关联;且 EADM 响应块存储(R)指示符,当被设置时,指示 EADM 响应块(ARSB)被存储。

[0311] 当 SCSW 的扩展的控制指示符和 SCSW 的扩展的状态字格式指示符被设置时, EADM 扩展控制字提供取决于模型的性质的额外信息,其描述了可能在 EADM 设施中存在的条件。

[0312] 此外,以下通道报告字(CRW)可针对 ADM 类型的子通道被报告;子通道安装的参数被初始化;子通道安装的参数被恢复;子通道可用;子通道事件信息待决。

[0313] 发生在 EADM 设施中的未经请求的事件和故障可被通道事件信息待决 CRW 报告。

[0314] 以上详细描述了扩展异步数据移动器设施,其用于在主存储器和存储级内存之间移动数据块,并执行存储级内存上的其他操作。在一个实施例中,使用 EADM 命令获得关于 EADM 设施和存储级内存的信息。特别地,由于存储级内存不是直接可被访问的,提供能力以确定存储级内存是否被分配,以及如果是这样,获取关于配置的信息。特别地,在一个例子中,提供能力以与控制程序(例如操作系统)交流是否存储级内存已被分配,以及如果是这

样,分配了多少以及在什么位置。确定可用的存储级内存的能力在此被称为发现,且发现功能的一个例子是由存储存储级内存(SCM)信息(SSI)命令提供的。

[0315] 存储存储级内存信息(SSI)命令被用于获取关于存储级内存以及扩展异步数据移动器设施的信息。SSI 命令返回以下从例如通道子系统获取的信息。

[0316] 1. EADM 设施的特征,包括 :

[0317] • 每个 AOB 的移动规格块(MSB)的最大计数。

[0318] • 每个 MSB 的最大块计数。

[0319] 2. 存储级内存的特征,包括 :

[0320] • SCM 增量大小。

[0321] • SCM 地址空间内的 SCM 地址增量列表。

[0322] • 取决于模型的最大 SCM 地址。

[0323] 执行存储 SCM 信息命令,其是同步的,不改变包含在通道子系统中的任何信息。

[0324] 关于 SSI 命令的进一步信息将参考图 5A-5D 进行描述。首先参考图 5A,在一个实施例中,用于存储 SCM 信息命令的命令请求块 500 包括例如 :

[0325] 长度 502 :指定命令请求快的长度的值。

[0326] 命令代码 504 :指定用于存储 SCM 信息命令的命令代码的值。

[0327] 格式(FMT) 506 :指定命令请求块的格式的值。

[0328] 继续令牌 508 :可请求从未完成的之前响应恢复的继续点的值。如果继续令牌的值是零,作出新的开始。如果继续令牌的值是非零且不被识别,作出新的开始。

[0329] SSI 命令的响应块 520 的一个实施例将参考图 5B 描述,且包括例如 :

[0330] 长度 522 :指示命令响应块的字节长度的值。

[0331] 响应代码 524 :描述尝试执行 SSI 命令的结果的值。响应代码值指定响应块的长度。例如,如果选定的响应代码被存储,长度指示 $96+N \times 16$ 字节,其中 N 是存储级内存地址列表项的数量,如以下描述的。在一个例子中,N 位于 $1 \leq N \leq 248$ 的范围。

[0332] 格式(FMT) 526 :指示命令响应块的格式的值。该字段的值是例如零。

[0333] RQ528 :响应修饰符值,如以下描述的 :

[0334] 不存在响应修饰。

[0335] 指定的继续令牌不被识别,且被处理为如同零已被指定。

[0336] 每个 MSB 最大块计数(MBC) 530 :指示可被用在移动规格块(MSB)的块计数字段中的最大值的值。

[0337] 最大 SCM 地址(MSA)532 :指示取决于模型的最大 SCM 地址的值。它是最高可寻址的 SCM 增量中的最后一个字节的 SCM 地址。

[0338] SCM 增量大小(IS) 534 :表示 SCM 地址列表中的每个 SCM 增量的大小的值,且例如是 2 的幂。

[0339] 最大 MSB 计数(MMC) 536 :指示可在 EADM 操作块(AOB) 中被指定的移动规格块(MSB)的最大计数的值。

[0340] 最大可配置 SCM 增量(MCI) 538 :可被配置到请求配置的 SCM 增量的最大数量的值。

[0341] 在一个实施例中,MCI 不超过 $2^{(64-IS)}$ 。例如,对于 16G 字节的 SCM 增量大小,

$MCI \leq 2^{(64-34)}$, 因为所有的 16G 字节的 SCM 增量可在 64 位地址的寻址约束中被寻址。而且, $((MCI+1) \times IS) - 1$ 不超过取决于模型的最大 SCM 地址。

[0342] 配置的 SCM 增量(NIC)的数量降低了当如下描述的成功的配置存储级内存命令被执行时可被指定的总大小(TS),这样 $TS \leq (MCI-NCI)$ 。但是,基于系统的整个容量和已对其他配置作出的分配,初始化状态中的 SCM 增量的数量可以能够或不能够完全满足请求,以便以其他方式有效地配置到请求的配置的 MCI 界限。

[0343] CPC 的总初始化 SCM 增量 540 :表示用于系统(例如,用于中央处理复合体(CPC))的初始化状态中的 SCM 增量的数量的值。如果系统被逻辑分区,这是分配给分区的 / 可由分区分配的总的可用增量的数量。

[0344] CPC 的总的未初始化 SCM 增量 542 :表示用于系统(例如用于 CPC)的未初始化状态中的 SCM 增量的数量。

[0345] SCM 测量块大小 514 :SCM 测量块的以字节表示的块大小(BS)的值。在一个例子中,它是 2 的幂,且最大的 SCM 测量块大小是例如 4096 字节。

[0346] SCM 资源部件 546 的最大数量 :作为 CPC 上的 SCM 资源部件(RP)(例如 I/O 适配器)的最大数量的值。每个 SCM 增量与 SCM 资源部件关联。每个 SCM 资源(例如一个或多个 I/O 适配器和一个或多个 SSD)包括一个或多个部件。SCM 资源部件的最大数量是例如 509。

[0347] 在一个例子中,术语“资源部件”被定义以用于获取与存储级内存有关的测量信息。每个 SCM 增量可被分配给多个适配器,且每个适配器具有与可告知的一些使用 / 测量数据。这样,返回的每个测量块被由增量标识符加上资源标识符组成的元组识别。

[0348] SCM 数据单元大小 548 :取决于模型的值,其指示包括在 SCM 数据单元中的字节的数量。在一个例子中,数据单元被定义为用于获得与存储级内存有关的测量信息。被报告的计数是数据单元而不是字节的计数。

[0349] 继续令牌 550 :取决于模型的值,通过该值 SSI 命令的随后发出可在由令牌表示的继续点继续。继续令牌的内容是取决于模型的。

[0350] 存储级内存地址列表 552 :SCM 地址列表的起点。当存储的响应代码是预定值时,多个 SCM 地址列表项(SALE)被存储(例如(长度 -96) /16 个 SALE 被存储)。

[0351] 在一个例子中,存储的 SALE 的数量取决于请求者的配置中的 SCM 增量的数量,当命令被执行时其每个的状态,以及通道子系统模型。零个或更多个 SALE 被存储,且在一个例子中,被存储的实际数量通过从响应块的大小(长度)减去 96 并将该结果除以 16 来确定。

[0352] 每个 SCM 地址列表项(SALE)表示一个 SCM 增量,其占用 SCM 地址的一个范围。由 SALE 表示的 SCM 增量的开始 SCM 地址包含在 SALE 中,且是对应的 SCM 增量的第一字节的 SCM 地址。在一个例子中,通过将以字节表示的 SCM 增量大小添加到起始 SCM 地址且随后减一来计算结束地址。这是 SCM 增量的最后一个字节的 SCM 地址。在一个实施例中,由 SALE 表示的存储级内存是连续一组 SCM 字节位置,其从自然 2IS 字节边界开始。

[0353] 当对应的 SCM 增量处于配置状态且空间在用于 SALE 的响应块中可用时,存储 SALE。如果响应块的 SCM 地址列表中的空间耗尽,一值被存储在继续令牌中且以特定的响应代码完成执行。

[0354] 两个或更多个 SALE 以升序顺序被存储在其 SCM 地址中。

[0355] 参考图 5C 描述的 SALE 的一个实施例。在一个实施例中,SALE552 包括例如 :

[0356] SCM 地址(SA)560 :作为 SCM 地址空间中的对应 SCM 增量的字节 0 的起始 SCM 地址的值,在由 SCM 增量大小(2IS 字节)确定的自然边界上对齐。

[0357] 持久性属性(P)562 :作为指示适用于 SCM 增量的当前持久规则的值。SCM 增量中的任何位置继承了持久性规则。可能的持久性规则包括 :

[0358] 规则 1 —当断电时保留数据。

[0359] 规则 2 —保留数据直到上电重置或 IML。

[0360] 操作状态 564 :指示由 SALE 表示的存储级内存增量的操作状态的值。仅当相关 SCM 增量处于配置状态时,操作状态是有效的。

[0361] 操作状态的例子包括 :

[0362] 可操作(Op):由 SALE 表示的存储级内存可用于所有的 I/O 操作。一经成功配置就进入该可操作状态,且当从临时或永久错误状态退出时可重新进入该可操作状态。

[0363] 临时错误(TE):由 SALE 表示的存储级内存不能用于任何 I/O 操作。数据状态是无效的,但是从可操作到临时错误的转变中的数据内容被保留。当对 SCM 增量的访问不存在时,从可操作状态进入临时错误状态。

[0364] 永久错误(PE):由 SALE 表示的存储级内存不可用于任何 I/O 操作。数据状态是无效的,且数据丢失。当识别了不可校正的错误条件时,从临时错误状态或可操作状态进入永久错误状态。

[0365] 当操作以在 EADM 响应块的异常标识符代码中设置永久错误指示完成时,至少该对应的 SCM 增量已进入到永久错误状态。但是,超过一个的 SCM 增量可能进入到永久错误状态。

[0366] 当 SCM 增量不在可操作状态时,引用增量中的位置的 I/O 操作以设置在 EADM 响应块的异常标识符代码中的临时或永久错误,识别扩展的异步数据移动设施检验。

[0367] 数据状态 566 :指示由 SALE 表示的存储级内存增量的内容的数据状态的值。当相关的 SCM 增量处于配置和可操作状态时,数据状态是有效的。

[0368] 示例性数据状态包括 :

[0369] 置零—SCM 增量的内容都是零。

[0370] 有效—SCM 增量的内容是所有成功的写类型操作的积累。未被写入的增量中的位置保持置零或不可预测。

[0371] 不可预测—在任何写类型 I/O 操作之前,SCM 增量的内容是未知的。在一个或多个写类型操作已被执行后,即使成功写入的位置导致 SCM 增量的数据状态变得有效,其他未写入位置的数据内容保持为不可预测。

[0372] 利用第一次成功写入,发生从置零或不可预测到有效状态的转变。由于写入的数据大小和目标 SCM 增量的大小之间的任何差异,到有效的变化不会描述尚未被写入的任何数据位置的实际情况。这样的还未被访问用于写入的位置仍然被有效地描述为置零或不可预测。

[0373] 等级 568 :指示由 SALE 表示的存储级内存增量的概念性质量的值。仅当 SCM 增量处于配置和可操作状态时,等级是有效的。零值意味着没有等级。指定范围中的非零值指示存在等级。在该例子中,1 的等级值是最高或最佳等级。15 的等级值是最低或最差等级。其他相同时,具有较高等级的 SCM 比具有较低等级的 SCM 更优先。

[0374] R570 :该字段指示 SCM 增量识别释放操作。以下行为是相关的：

[0375] 1. 释放的块在被读取前先被写入,否则如果读取在写入之前,识别读取操作错误。对于这样的错误,SCM 增量保持在可操作状态中。

[0376] 2. 一旦进行初始配置,数据状态为置零。

[0377] 3. 程序可执行特殊操作,被称为释放,其将指定的块置于释放条件中。

[0378] 资源 ID572 :非零值是提供由 SALE 表示的 SCM 增量的资源的资源标识符(RID)。当 RID 是零时,没有资源 ID 被指示。在一个特定例子中,RID 表示提供用于 SCM 增量的存储的同样多的适配器和 SSD。因为可应用特定的 RAID 算法,或通过允许跨过多个适配器 /SSD 的并行 I/O 操作来条带化(striping)以改善性能,RID 可表示复合实体。

[0379] 关于存储级内存的配置状态的进一步细节,以及存储级内存地址列表的操作和数据状态将在以下描述。

[0380] 首先,参考图 6A,描述配置状态和导致这些状态内的转变的事件 / 动作。如图所示,SCM 状态是配置、备用和保留。SCM 可由保留被置于备用,随后由备用到配置。从配置,SCM 可被解除配置并进入保留状态。

[0381] 参考图 6B,示出了操作状态和在这些状态内导致转变的事件。SCM 增量将是处于备用状态以便被配置,且在配置动作成功完成后处于操作状态。到置零状态中的 SCM 增量的第一写入将其移动到有效状态。对于未被指示为具有规则 1 持久性的 SCM 增量的居间的断电和上电将 SCM 增量移动到不可预测状态。

[0382] 错误(E)可引起到临时错误(TE)状态或永久错误(PE)状态的转变,这取决于错误的取决于模型的特定细节。连接的获取(A)可引起从临时错误状态到操作(Op)状态的转变。可发生 SCM 增量的解除配置,而不管其操作状态。

[0383] 图 6B 也示出了当在可操作状态时取决于如何进入了可操作状态的数据状态。当被配置且处于可操作状态时,数据状态时有效的,并被施加给对应的 SCM 增量。有效数据状态是置零、不可预测和有效。以下是到可操作状态的各种进入时可能的数据状态:

[0384] . 从备用—置零(z)

[0385] . 从临时错误—有效(v)

[0386] . 从永久错误—不可预测(u) 或置零(v)

[0387] . 从可操作—有效(V) —第一次写入

[0388] . 从可操作—不可预测(u) —功率循环且持久性不是规则 1。

[0389] 当未处于操作状态时,数据状态是无效。

[0390] 当第一次被配置且在第一次写入之前,SCM 增量的数据处于置零状态,意味着其内容都是零。

[0391] 尽管当移动到或处于临时错误状态时,SCM 增量的数据内容不被改变,增量是不可访问的。因此,说数据是无效的可能是描述性的,但由于缺乏程序可访问性不会过于有意义。因此,数据状态在该场景下是无效。而且,基于引起从可操作到临时错误的转变的错误,如果数据整体性被影响,进入永久错误状态,数据状态保持为无效,且数据丢失。如果并行修复可将 SCM 增量从永久错误状态移动到可操作状态而不会即被解除配置,又随后再次配置,原始数据仍然丢失,且数据状态是不可预测还是置零,这取决于模型。

[0392] SCM 增量的持久性及其 RAS (可靠性、可访问性和可服务性)特点也可确定从有效

到不可预测数据状态的变化。如果持久性被超过,可预期从有效到不可预测的数据状态转变。

[0393] 随着第一成功写入,发生从置零或不可预测的数据状态到有效数据状态的变化。由于写入的数据大小和目标 SCM 增量大小之间的任何差别,到有效的变化不会描述任何尚未被写入的数据位置的实际情况。这样的位置,在首次写入访问之间,仍然被有效地描述为置零或不可预测。

[0394] 在 SCM 增量被配置后,当任何一个或多个可在存储 SCM 信息命令的响应中观察到的事件发生,使未经请求的通知待决。例子如下:

[0395] 1. 操作状态从可操作变化到临时错误或永久错误,但不会在失败操作中被报告。

[0396] 2. 操作状态从临时错误变化到可操作。

[0397] 3. 操作状态从临时错误变化到永久错误。

[0398] 4. 等级变化。

[0399] 其中未经请求的通知没有变得待决的例子包括以下:

[0400] 1. 数据状态从置零或不可预测改变到有效。

[0401] 2. 配置存储级内存命令完成。

[0402] 3. 解除配置存储级内存命令完成。

[0403] 当通知待决时,程序观察到通知并可发出存储 SCM 信息命令以获取信息。存储 SCM 信息命令也可在其他时间被发出,在所述时间程序将需要关于 SCM 和 / 或 SALE 的信息。

[0404] 在一个例子中,通知包括被发出给程序的机器检查中断,其具有指示事件报告的对应 CRW。程序发出 CHSC 存储事件信息命令并获取具有内容代码的响应块,其信号通知存储级内存变化通知。

[0405] 在一个实施例中,存储 SCM 信息(SSI)命令是由程序(例如操作系统)发出的通道子系统命令以获取关于存储级内存和 / 或 SCM 地址列表项的信息。在一个例子中,程序发出通道子系统调用指令,且 SSI 命令在该指令的命令块中被指示,其被发出到通道子系统。命令在通道子系统处被执行,且响应在响应块被返回,响应块是 4K 字节控制块的剩余部分(即,请求的信息存储在为响应块指定的主存储器区域中)。参考图 5D 描述关于该命令的操作的进一步细节。

[0406] 首先,在步骤 580 程序生成以上指示的请求块以请求存储 SCM 信息命令。在步骤 582 请求块被通道子系统获得,且进行一个或多个有效性检验,以检验请求快的有效性(例如,有效长度字段、有效命令请求块格式、命令已安装等)。如果在询问 584 请求不是有效的,则在步骤 586 指示问题的响应代码被放置在响应块中,且在步骤 592 响应块被返回。

[0407] 但是,如果在询问 584 请求是有效的,则在步骤 588 通道子系统获取来自机器(例如处理器等)的信息,并在步骤 590 填充响应块。在步骤 592 返回响应块。例如,信息被包含在机器的非易失性存储器中,且在系统初始化期间,被固件加载到仅可被固件访问的主存储器中。通过将信息从仅可被固件访问的主存储器读取,通道子系统(即该例子中的固件)获取信息,并填充响应块。

[0408] 响应于接收到关于存储级内存的信息或没有接收到该信息,可做出决定改变存储级内存的配置。该决定可手动做出或者被程序或其他实体自动做出。如以下描述的,可通过添加增量或删除增量来改变配置。

[0409] 在一个例子中,为了配置存储级内存,使用配置存储级内存命令。该命令请求将从系统的可用池被配置的存储级内存的量。该量被指定为大小,被编码为 SCM 增量的计数。

[0410] 除非另外清楚说明,被用来满足请求的 SCM 增量的数量处于初始化状态。如果请求的 SCM 增量的数量使得最大可配置的 SCM 增量界限被超过,提供特定的响应代码。

[0411] 每个增量的内容是具有有效 CBC 的零。与每个配置的 SCM 增量关联的可应用持久规则是由例如手动控制设置的。

[0412] 用于配置存储级内存命令的命令请求块的一个实施例在图 7A 中示出。在一个例子中,配置存储级内存请求块 700 包括 :

[0413] 长度 702 :指示命令请求块长度的值。

[0414] 命令代码 704 :指示用于配置存储级内存命令的命令代码的值。

[0415] 格式(FMT) 706 :指示命令请求块的格式的值。

[0416] 总大小(TS) 708 :指示被编码为 SCM 增量的计数的、请求的存储级内存的大小的值。已被配置的 SMC 增量的计数加上 TS 不应超过最大可配置 SCM 增量(MCI) 界限。如果初始化状态中的 SCM 增量的数量小于指定的总大小,提供特定的响应代码。

[0417] 异步完成相关器(ACC) 710 :在以下描述通知响应的异步完成通知字段中返回的值。相关器用于恢复启动请求的原始线程。

[0418] 用于配置存储级内存命令的命令响应块的一个实施例在图 7B 中描述。在一个实施例中,命令响应块 730 包括 :

[0419] 长度 732 :指示命令响应块的长度的值。

[0420] 响应代码 734 :描述尝试执行配置存储级内存命令的结果的值。

[0421] 如果定义的响应代码被存储在响应代码字段中,异步过程被启动以完成命令的处理。如果定义的代码之外的响应代码被存储在响应代码字段中,没有 SCM 增量被配置,没有异步过程被启动,且不做出随后的通知。异步过程的完成在通知响应中被指示。

[0422] 格式(FMT) 746 :指示命令响应块的格式的值。

[0423] 配置存储级内存命令由程序发出以请求将被配置到 SCM 地址空间中的存储级内存的量。用于配置 SCM 的逻辑的一个实施例将参考图 7C 进行描述。

[0424] 开始,在步骤 740 程序发出通道子系统调用指令,其包括配置 SCM 命令。在步骤 742,配置 SCM 命令的请求块由通道子系统获得,且在步骤 744 通道子系统试图执行命令。如果在询问 746 尝试执行命令产生不指示成功的响应代码,则在步骤 748 响应代码被放置在配置 SCM 命令的响应块中,且在步骤 750 响应块被返回。

[0425] 如果在询问 746 成功的响应代码被指示,则在步骤 752 响应代码被放置在响应块中,且在步骤 754 响应块被返回。在该例子中,成功的响应代码指示请求快的长度字段是有效的;命令在系统中是可用的;命令请求块具有有效格式;通道子系统能执行命令(即,不忙);请求的总大小不超过请求的配置的最大可配置 SCM 增量界限;且请求的总大小不超过初始化状态中的 SCM 增量的数量。

[0426] 此外,在步骤 756 启动完成配置的异步过程。参考图 7D 描述关于该处理的进一步细节。

[0427] 在一个实施例中,在步骤 760 异步处理执行配置以分配一个或多个增量。例如,对于每个配置的 SCM 增量,内部控制被改变以允许新配置的增量可被对该分区的 I/O 移动请

求访问。特别地,响应于通道子系统接收 CHSC 配置命令,通道子系统的固件检查内部表以确认存在足够的增量以满足请求,并确保请求不超过用于配置的最大可配置 SCM 增量。如果请求是有效的,固件更新一个或多个表以分配增量到配置,并将增量放置到用于配置的可操作状态。增量随后可被来自配置的 I/O 移动请求(如上描述的)访问。在步骤 762 异步过程的完成在通知响应中被指示。

[0428] 用于配置存储级内存命令的通知响应数据在存储事件信息(SEI)命令的响应块中被返回。参考图 8A 描述用于配置存储级内存命令的通知响应块的格式的一个实施例。

[0429] 在一个实施例中,配置存储级内存命令的通知响应块 800 包括:

[0430] 长度 802 :指示命令响应块的长度的值。

[0431] 响应代码 804 :描述试图执行存储事件信息 CHSC 命令的结果的值。

[0432] 格式(FMT) 806 :指示命令响应块的格式的值。

[0433] 通知类型 808 :指示这是一个 EADM 相关通知的值。

[0434] P810 :当被设置时,指定除了响应于该 CHSC 命令被提供的信息外,通道子系统具有待决事件信息。

[0435] V812 :当被设置时,指定通道子系统已识别溢出条件且事件信息已被丢失。

[0436] 内容代码 814 :指示这是对由配置存储级内存命令请求启动的异步过程的执行完成的响应的值。

[0437] 次级响应代码 816 :进一步描述尝试执行配置 SCM 命令的结果的值。

[0438] 当次级响应代码是指定值时,如开始请求的存储级内存的完全量已被配置。否则,响应代码可被提供,其指示例如无效的长度字段、配置 SCM 命令未安装、配置 SCM 命令块具有无效格式、总请求大小将超过 MCI 界限、请求的总大小超过初始化状态中的 SCM 增量的数量、通道子系统忙。

[0439] 异步完成相关器(ACC) 818 :在对应的命令请求块中初始指定的值。

[0440] 参考图 8B-8C 描述用于返回配置存储级内存命令的通知响应块的存储事件信息命令的一个实施例。

[0441] 存储事件信息命令用来存储已被通道子系统变得待决的事件信息。通常,该命令作为已接收了事件信息待决通道报告的程序的结果而被执行。

[0442] 存储事件信息命令的执行可改变包含在通道子系统中的信息。存储事件信息命令被同步执行。

[0443] 参考图 8B 描述用于存储事件信息命令的命令请求块的一个实施例。在一个例子中,请求块 830 包括:

[0444] 长度 832 :该字段指定命令请求块的长度。

[0445] 命令代码 834 :该字段指定存储事件信息命令。

[0446] 格式(FMT) 836 :指定命令请求块的格式的值。

[0447] 通知类型选择掩码(NTSM) 838 :掩码,其每个位位置对应于相同数字值的逻辑处理器选择器(LPS)值。在一个实施例中,位 0 被忽略并被假设是 1。当从 1 开始的指定范围中的位位置是零时,对应于该位位置的通知类型没有被存储在响应块中,且如果被识别为待决,则被丢弃。当这样的位是 1 时,对应于位位置的通知类型可被存储在响应块中。

[0448] 在一个实施例中,参考图 8C,用于存储事件信息命令的响应块 850 被如下描述:

[0449] 长度 852 :指定命令响应块的初始长度的值。完成长度取决于作为试图执行存储事件信息命令的结果而被存储的响应代码。

[0450] 如果指示成功的代码之外的响应代码被存储在响应代码字段中, 没有信息被存储在响应数据区域中。

[0451] 如果指示成功的响应代码被存储在响应代码字段中, 事件信息被存储在响应数据区域中。

[0452] 响应代码 854 :描述试图执行存储事件信息命令的结果的值。

[0453] 对于存储事件信息命令, 响应数据区域包含固定长度部分和可变长度部分。

[0454] 对于指定的格式响应, 当 NT 是非零时, 格式取决于特定通知类型, 且取决于字段的内容代码的格式取决于特定通知类型和内容代码(CC) 字段的组合。

[0455] 格式(FMT) 856: 指示命令响应块的格式的值。

[0456] 通知类型(NT) 858 :指示通知类型(NT) 的值。提供用于配置 SCM 命令的特定值。

[0457] F 标志 860 :当被设置时, 指定通道子系统除了响应于该 CHSC 命令而被提供的信息, 具有待决的事件信息。程序可通过再次执行存储事件信息命令获得额外的信息。当未被设置时, 该标志指定通道子系统不具有额外的待决事件信息。

[0458] V 标志 862 :当被设置时, 指定通道子系统已识别了溢出状况且事件信息已被丢失。当没有被包含在响应数据区域中的事件信息是最近的待决信息时, 溢出状况被识别。溢出不影响包含在响应数据区域中的信息。

[0459] 内容代码(CC) 864 :描述包含在响应数据区域中的信息的类型的值。在一个例子中, 该值指示存储级内存变化通知, 其中一个或多个 SCM 增量已改变了状况或状态。

[0460] 内容代码依赖字段 866 :该字段包括关于事件的额外信息。

[0461] 配置变化的成功通知可促使程序发出存储 SCM 信息命令以获得关于配置的细节。

[0462] 除了增加存储级内存, 存储级内存也可被减少。解除配置存储级内存命令请求存储级内存的量从请求配置的 SCM 地址空间移除。将被解除配置的 SCM 增量将处于配置状态。

[0463] 如在此描述的, 将被解除配置的 SCM 增量在 SCM 增量请求列表中被指定。一个或多个连续的 SCM 增量可在 SCM 增量请求列表项(SIRLE) 中被指定。单独的 SIRLE 可被指定用于增量(也称为区段(extent)) 的每个列表, 其不与任何其他增量列表连续。

[0464] 不管持久规则, SCM 增量的成功解除配置将增量放到未初始化状态。当置零完成时, SCM 增量从未初始化状态转变到初始化状态。

[0465] 用于解除配置存储级内存命令的命令请求块的一个实施例在图 9A 中示出。用于解除配置存储级内存命令的命令请求块 900 包括例如 :

[0466] 长度 902 :指定命令请求块的长度的值。在一个实施例中, 该长度是 $32 + (N \times 16)$ 字节, 其中 N 是 SCM 增量请求列表项(SIRLE) 的计数。在该例子中长度可被 16 整除, 且处于 $(32 + 1 \times 16) \leq L_1 \leq (32 + 253 \times 16)$ 的范围。

[0467] 命令代码 904 :指定用于解除配置存储级内存命令的命令代码的值。

[0468] 格式(FMT) 906 :指定命令请求快的格式的值。

[0469] 异步完成相关器(ACC) 908 :在如上所述的异步完成通知中返回的值。

[0470] SCM 增量请求列表 910 :该字段包括 SCM 增量请求列表(SIRL)。SCM 增量请求列表包括一个或多个项(SIRLE)。SIRL 的长度是从长度字段的值确定的。

[0471] SCM 增量请求列表项(SIRLE)指定存储级内存(例如增量列表)的所指定区段的大小和位置。区段或 SCM 区段是存储级内存的指定大小。

[0472] 参考图 9B,在一个例子中, SIRLE920 包括 :

[0473] 总大小(TS)922 :指定编码为 SCM 增量的计数的、将被解除配置的存储级内存的大小。

[0474] 开始 SCM 地址(SA)924 :该值是 SCM 地址,且是 SCM 地址空间中的位置,从其移除由 SIRLE 解除配置的第一个或仅有的 SCM 增量。在该例子中,将构成第一 SCM 增量内的偏移的最低位位置被忽略,且被假设是零。

[0475] 当总大小大于 1 时,超过第一增量的每个额外的 SCM 增量位于可被 SCM 增量大小整除的 SCM 地址处,其包含配置的 SCM 增量,且其位置与先前的 SCM 增量的最后一个字节连续。换句话说,位于下一个连续位置。

[0476] 如果由开始位置和总大小的组合描述的空间不是完全充满配置的 SCM 增量,提供指定的响应代码,没有 SCM 增量被配置,没有异步过程被启动,且没有随后的通知发生。

[0477] 一旦成功完成,每个解除配置的 SCM 增量进入到保留状态,且随后在被放置到备用状态前被置零。

[0478] 用于解除配置存储级内存命令的命令响应块在图 9C 中示出。在一个实施例中,命令响应块 950 包括 :

[0479] 长度 952 :指示命令响应块的长度的值。

[0480] 响应代码 954 :描述试图执行解除配置存储级内存命令的结果的值。

[0481] 如果指定值的响应代码被存储在响应代码字段中,异步过程被启动以完成命令的处理。如果指定值以外的响应代码被存储在响应代码字段中,没有 SCM 增量被解除配置,没有异步过程被启动,且没有随后的通知被做出。异步过程的完成在通知响应中被指示。

[0482] 在解除配置存储级内存命令的同步部分以指定的响应代码完成前,SCM 增量请求列表中的所有项被检查以确保所有指定的 SCM 增量处于配置状态。

[0483] 格式(FMT)956 :指示命令响应块的格式的值。参考图 9D-9E 描述与解除配置 SCM 命令关联的逻辑的一个实施例。

[0484] 开始,在步骤 970 程序发出包括解除配置 SCM 命令的通道子系统调用指令。在步骤 972 解除配置 SCM 命令的请求块被通道子系统获取,且在步骤 974 通道子系统试图执行命令。如果在询问 976 尝试执行命令产生不指示成功的响应代码,则在步骤 978 响应代码被放置在解除配置 SCM 命令的响应块中,且在步骤 980 响应块被返回。

[0485] 如果在询问 976 成功的响应代码被指示,则在步骤 982 响应代码被放置在响应块中,且在步骤 984 响应块被返回。在该例子中,成功的响应代码指示请求块的长度是有效的;命令在系统中可用;命令请求块具有有效格式;通道子系统能执行命令(即,不忙);且 SCM 增量原始地处于配置状态。

[0486] 此外,在步骤 986 完成解除配置的异步处理被启动。关于该处理的进一步细节将参考图 9E 进行描述。

[0487] 在一个实施例中,在步骤 900 异步处理执行解除配置。例如,一个或多个增量被解除分配。SCM 增量从配置状态移动到保留状态。一旦进入到保留状态,接着发生置零处理,且当完成时,SCM 增量转变到备用状态。在步骤 992 异步过程的完成在通知响应中被指示。

[0488] 用于解除配置存储级内存命令的通知响应数据在存储事件信息(SEI) CHSC 命令的响应块中被返回。参考图 8A 描述该响应块的一个例子。但是,该例子中的内容代码指示这是对于由解除配置存储级内存命令请求启动的异步过程的执行完成的响应。类似地,次级响应代码还描述了试图执行解除配置存储级内存命令的结果。

[0489] 在又一个实施例中,存储增量的分配和解除分配可经由呈现给用户的面板而被请求。例如,服务元件被用来提供图形界面,通过该图形界面用户可对系统指定参数。对于存储级内存,称为存储级内存分配面板的面板允许用户执行以下操作:

[0490] 1. 指定用于给定配置的最大可配置增量(MCI);

[0491] 2. 分配增量给配置;

[0492] 3. 从配置解除分配增量。

[0493] 面板也允许查看配置增量分配和 MCI,以及可用、不可用和未初始化池中的增量的数量。当由于 SE 处的动作,增量分配改变或当一个池的大小改变时,通知被发送到配置。

[0494] 以上详细描述了管理存储级内存的设施。它提供了抽象以允许程序访问存储器,而不需要对存储器的特定知识。根据本发明的一个或多个方面,提供能力以在主存储器和 SCM 之间移动数据;以清除或释放 SCM;以配置或解除配置 SCM;以及以发现 SCM 的配置。也提供了其他的能力。

[0495] 在一个实施例中,存储级内存对用户级别程序被呈现为平坦内存空间,独立于其跨过多个设备和 I/O 适配器的物理实现。

[0496] 关于通道子系统和 / 或 ADM 设施的细节在以下文献中被描述 :Antognini 等人的、于 1994 年 12 月 27 日发布的、标题为“Method and Means for Enabling Virtual Addressing Control By Software Users Over A Hardware Page Transfer Control Entity”的美国专利号 5377337;Brent 等人的、于 1995 年 8 月 15 日发布的、标题为“Asynchronous Co-Processor Data Mover Method and Means”的美国序列号 544802;以及 1996 年 6 月 11 日发布的、标题为“Method and System for Pipelining the Processing of Channel Command Words”的美国专利号 5526484,这些文献中的每一个通过全文引用结合于此。

[0497] 此外,关于通道子系统和与其相关的指令的进一步的信息(用于 **z/Architecture®** 的特定实现)被提供如下:

[0498] 输入 / 输出(I/O)

[0499] 术语“输入”和“输出”被用来描述 I/O 设备和主存储器之间的数据传送。涉及这种类型的传送的操作被称为 I/O 操作。用于控制 I/O 操作的设施被统称为通道子系统。(I/O 设备及其控制单元附着于通道子系统。)

[0500] 通道子系统

[0501] 通道子系统引导 I/O 设备和主存储器之间的信息流。它解除了 CPU 直接与 I/O 设备通信的任务,并允许数据处理与 I/O 处理并行进行。通道子系统使用一个或多个通道路径作为通信链路,以管理到或来自 I/O 设备的信息流。作为 I/O 处理的一部分,通道子系统也通过测试通道路径可用性执行路径管理操作,选择可用的通道路径,并启动由设备执行的 I/O 操作。

[0502] 在通道子系统内的是子通道。一个子通道被提供并专用于通过通道子系统可由程

序访问的每个 I/O 设备。

[0503] 多个子通道组设施是可选设施。当其被安装时，子通道被分区为多个子通道组，且每个子通道组可提供一个专用的子通道给 I/O 设备。根据使用的模型和界面，一些 I/O 设备可仅被允许经由特定的子通道组来访问。

[0504] 每个子通道提供了关于相关 I/O 设备及其到通道子系统的连接的信息。子通道也提供关于 I/O 操作和涉及相关 I/O 设备的其他功能的信息。子通道是通道子系统提供关于相关 I/O 设备的信息给 CPU 的手段，CPU 通过执行 I/O 指令获取该信息。所提供的子通道的实际数量取决于模型和配置；每个子通道组中的最大可寻址性是 0-65535。

[0505] I/O 设备通过控制单元经由通道路径被附加到通道子系统。控制单元可通过多于一个通道路径被附加到通道子系统，且 I/O 设备可被附加到多于一个的控制单元。总之，单个 I/O 设备可经由子通道通过多达 8 条不同的通道路径被通道子系统访问，这取决于模型和配置。通道子系统提供的通道路径的总数取决于模型和配置；最大可寻址性是 0-255。

[0506] 通道子系统的性能取决于其使用和其被实施于其中的系统模型。通道路径被提供有不同的数据传送能力，且被设计为仅以特定速率传送数据的 I/O 设备（例如磁带单元或盘存储器）可在能允许至少该数据速率的通道路径上操作。

[0507] 通道子系统包含用于 I/O 操作的控制的共同设施。当这些设施以被特别设计为控制 I/O 设备的单独的自主设备的形式提供时，I/O 操作完全与 CPU 中的活动重叠。在 I/O 操作期间由通道子系统要求的仅有的主存储器循环是传送数据和控制信息到或来自主存储器中的最终位置的所需要的那些循环，以及当子通道被实施为不可寻址的主存储器的一部分时，由通道子系统访问子通道所需的那些循环。这些循环不会延迟 CPU 程序，除了当 CPU 和通道子系统同时试图引用相同的主存储器区域时。

[0508] 子通道组

[0509] 当多个子通道组设施被安装时，子通道被分区为多个子通道组。可以有多达 4 个子通道组，每个被子通道组标识符（SSID）标识。当多个子通道组设施未被安装时，仅有一个子通道组，其具有为零的 SSID。当多个子通道组设施未被使能时，仅子通道组零对于程序是可见的。

[0510] 子通道

[0511] 子通道向程序提供了设备的逻辑外表，并包含用于维持单个 I/O 操作所需的信息。子通道由内部存储构成，该内部存储包含的信息形式为通道—程序指定、通道—路径标识符、设备号、计数、状态指示以及 I/O 中断子类代码，以及关于路径可用性和待决或正被执行的功能的信息。I/O 操作通过执行 I/O 指令针对设备启动，所述 I/O 指令指定与设备关联的子通道。

[0512] 每个设备可通过每个通道子系统中的一个子通道访问，在安装时的配置期间它被分配给该子通道。设备可以是物理可识别单元，或可被放置到控制单元的内部。例如，在特定的盘存储设备中，用于检索数据的每个致动器被认为是一个设备。在所有的情况下，从通道子系统的观点来看，设备是与一个子通道唯一关联的一个实体，且对通道子系统通过使用通信协议进行的选择进行响应，所述通信协议是为通道路径的类型定义的，通过该通道路径可访问该设备。

[0513] 在一些模型上，子通道按块被提供。在这些模型上，比附加设备更多的子通道可被

提供。被提供但不具有分配给其的设备的子通道被通道子系统使用以执行任何功能,且通过将关联的设备号有效位作为零存储在子通道的子通道信息块中而被指示。

[0514] 由通道子系统提供的子通道的数量与被提供给相关设备的通道路径的数量无关。例如,可通过替代通道路径被访问的设备仍然被单个子通道表示。当子通道组设施被安装时,通过使用 16 位的二进制子通道号和二位的 SSID,每个子通道被寻址。

[0515] 在子通道处的 I/O 处理已被 START SUBCHANNEL 的执行请求之后,CPU 被释放用于其他工作,且通道子系统装配或拆开数据并同步 I/O 设备和主存储器之间的数据字节的传送。为了完成这一点,通道子系统维护并更新地址,以及描述主存储器中的数据的目的或源的计数。类似地,当 I/O 设备提供应当引起程序注意的信号时,通道子系统将信号转变为状态信息,并将信息存储在子通道中,在此它可被程序检索。

[0516] 输入 / 输出设备的附加

[0517] 通道路径

[0518] 通道子系统通过通道子系统和控制单元之间的通道路径与 I/O 设备通信。控制单元可通过多于一个的通道路径被通道子系统访问。类似地,I/O 设备可通过多于一个的控制单元被通道子系统访问,每个控制单元具有一个或多个到通道子系统的通道路径。

[0519] 通过被配置到子通道的多个通道路径附加到通道子系统的设备可被通道子系统使用任何可用的通道路径访问。类似地,具有动态重连特征并在多路径模式中操作的设备可被初始化以操作,使得当逻辑重连到通道子系统以继续 I/O 操作链时,设备可选择被配置到子通道的任何可用的通道路径。

[0520] 通道子系统可包含多于一种类型的通道路径。通道子系统使用的通道路径类型的例子是 ESCON I/O 接口、FICON I/O 接口、FICON 转换的 I/O 接口以及 IBM System/360 和 System/370I/O 接口。术语“串行 I/O 接口”被用来指 ESCON I/O 接口、FICON I/O 接口、FICON 转换的 I/O 接口。术语“并行 I/O 接口”被用来指 IBM System/360 和 System/370I/O 接口。

[0521] ESCON I/O 接口在系统库公开 IBM Enterprise Systems Architecture/390ESCON I/O Interface, SA22-7202 中描述,其通过全文引用结合于此。

[0522] FICON I/O 接口在 ANSI 标准文档 Fibre Channel-Single-Byte Command Code Sets-2(FC-SB-2) 中被描述。

[0523] IBM System/360 和 System/370I/O 接口系统库公开 IBM System/360 and System/370I/O Interface Channel to Control Unit OEMI, GA22-6974 中被描述,其通过全文引用结合于此。

[0524] 根据通道路径的类型、通道路径提供的设施、以及 I/O 设备, I/O 操作可发生在三种模式中的一种中 :帧复用模式、猝发模式和字节复用模式。

[0525] 在帧复用模式中,在通道程序的执行期间 I/O 设备可保持逻辑连接到通道路径。能够在帧复用模式中操作的通道路径的设施可被若干个并行操作的 I/O 设备共享。在该模式中,完成 I/O 操作所需要的信息被分为帧,所述帧可与来自用于其他 I/O 设备的 I/O 操作的帧交错。在该期间,多个 I/O 设备被认为逻辑连接到通道路径。

[0526] 在猝发模式中, I/O 设备独占通道路径并保持逻辑连接到通道路径以用于信息猝发的传送。在猝发被传送期间,没有设备可在通道路径上通信。猝发可由若干字节、整块

数据、具有相关控制和状态信息的块序列(块长度可以是零)或独占通道路径的状态信息组成。能在猝发模式中操作的通道路径的设施可被若干个并行操作的 I/O 设备共享。

[0527] 一些通道路径可忍受在猝发模式操作期间大约半分钟的数据传送缺失,诸如当磁带上的长空隙被读取时发生的。当数据传送缺失超过规定界限时,可指示设备故障。

[0528] 在字节复用模式中,I/O 设备仅可在较短时间间隔保持逻辑连接到通道路径。能在字节复用模式中操作的通道路径的设施可被若干并行操作的 I/O 设备共享。在该模式中,所有的 I/O 操作被分为一些短的时间间隔,在时间间隔期间仅一段信息通过通道路径被传送。在这样的间隔期间,仅一个设备及其相关子通道逻辑连接到通道路径。与多个 I/O 设备的并行操作关联的间隔响应于来自设备的要求而被排序。与子通道关联的通道子系统设施仅在传送信息段所需的时间为任一个操作实施其控制。段可由单个字节的数据、若干字节的数据、来自设备的状态报告或用于启动新操作的控制序列组成。

[0529] 通常,具有高数据传送速率要求的设备使用帧复用模式中的通道路径操作,较慢的设备在猝发模式中操作,且最慢的设备在字节复用模式中操作。一些控制单元具有用于设置希望的操作模式的手动开关。

[0530] 发生在并行 I/O 接口类型的通道路径的 I/O 操作可发生在猝发模式或字节复用模式中,这取决于通道路径和 I/O 设备提供的设施。为了改善的性能,一些通道路径和控制单元可被提供有用于高速传送和数据流式传输的设施。参考系统库公开 IBM System/360and System/370I/O Interface Channel to Control Unit OEMI, GA22-6974 来获得这两个设施的描述,该公开通过全文引用结合于此。

[0531] 发生在串行 I/O 接口类型的通道路径的 I/O 操作可发生在帧复用模式或猝发模式中。为了改善的性能,附加到串行 I/O 接口的一些控制单元提供能力以与呈现单元检验状态并行地提供感测数据(sense data)给程序,如果被程序允许这么做的话。

[0532] 根据控制单元或通道子系统,通过子通道访问设备可被限制为单通道路径类型。

[0533] 上述模式和特点仅影响用于在通道路径传送信息的协议以及传输速率。CPU 或通道路程不会观察到针对这些程序被执行的方式的影响。

[0534] 控制单元

[0535] 控制单元提供操作和控制 I/O 设备所需的逻辑能力,并适应性调整每个设备的特点,这样它可响应于通道子系统提供的控制的标准形式。

[0536] 控制单元和通道子系统之间的通信发生在通道路径上。控制单元接收来自通道子系统的控制信号,控制经过通道路径的数据传送的时间,并提供关于设备状态的指示。

[0537] 附加到控制单元的 I/O 设备可被设计为仅执行特定的有限操作,或者它可执行许多不同的操作。典型的操作移动记录介质并记录数据。为了完成其操作,设备需要其设备类型特有的详细的信号序列。控制单元解码从通道子系统接受的命令,将其解释为用于特定类型的设备,并提供执行操作所需要的信号序列。

[0538] 控制单元可被单独放置,或其可被物理地和逻辑地与 I/O 设备、通道子系统或 CPU 集成。在大多数机电设备的情况下,由于控制单元和设备要求的设备类型差异,定义良好的接口存在于设备和控制单元之间。这些机电设备通常是这样的类型,其中在一时间仅要求附加到控制单元的组中的一个设备传送数据(例如磁带单元或盘访问装制),且控制单元在若干个 I/O 设备之间共享。另一方面,在诸如通道到通道适配器的一些电子 I/O 设备中,控

制单元不具有其自己的身份。

[0539] 从程序员的角度,由控制单元执行的大部分功能可与 I/O 设备执行的功能合并。因此,通常在该本说明书中没有特别提到控制单元功能;I/O 操作的执行被描述,就好比 I/O 设备直接与通道子系统通信。仅当强调控制单元执行的功能或描述在若干设备间共享控制单元会如何影响 I/O 操作的执行时,才参考控制单元。

[0540] I/O 设备

[0541] 输入 / 输出(I/O)设备提供了外部存储器、数据处理系统之间的通信手段或系统及其环境之间的通信手段。I/O 设备包括这样的设备,诸如磁带单元、直接访问存储设备(例如盘)、显示单元、打字机—键盘设备、打印机、远程处理设备和基于传感器的设备。I/O 设备可以是物理上单独的设备,或者其可与其他 I/O 设备共享设备。

[0542] 大多数类型的 I/O 设备,诸如打印机或磁带设备,使用外部介质,且这些设备是物理可区分和可识别的。其他类型是完全电子的,且不直接处理物理记录介质。例如,通道到通道适配器提供两个通道路径之间的数据传送,且数据从来不到达主存储器外的物理记录介质。类似地,通信控制器可处理数据处理系统和远程站之间的信息传输,且其输入和输出是传输线上的信号。

[0543] 在最简单的情况下,I/O 设备被附加到一个控制单元,且可从一个通道路径被访问。开关设备通过控制单元间切换设备以及通过通道路径间切换控制单元,可用于使得一些设备可从两条或更多条通道路径被访问。这样的开关设备提供多条路径,通过所述路径 I/O 设备可被访问。多条到 I/O 设备的通道路径被提供以改善系统内的性能或 I/O 可用性或这两者。到设备的多条通道路径的管理是在通道子系统和设备的控制下,但是通道路径可被程序间接控制。

[0544] I/O 寻址

[0545] 由通道子系统提供 4 种不同类型的 I/O 寻址,以用于各种组件的必要寻址:通道路径标识符、子通道号、设备号,以及虽然对程序来说不可见,取决于通道路径类型的地址。当多子通道组设施已安装时,子通道组标识符(SSID)也在 I/O 寻址中被使用。

[0546] 子通道组标识符

[0547] 子通道组标识符(SSID)是被分配给每个提供的子通道组的二位值。

[0548] 通道路径标识符

[0549] 通道路径标识符(CHPID)是被分配给系统的每个安装的通道路径的系统唯一的 8 位值。CHPID 被用来寻址通道路径。CHPID 被 RESET CHANNEL PATH(重置通道路径)的第二操作数地址指定,且被用来指定将被重置的通道路径。当 STORE SUBCHANNEL(存储子通道)被执行时,设备经其可被访问的通道路径在通道信息块(SCHIB)中被识别,每个设备由其相关 CHPID 识别。当需要识别特定通道路径时,CHPID 也可被用在操作符消息中。系统模型可提供多达 256 条通道路径。通道路径的最大数量和向通道路径分配 CHPID 取决于系统模型。

[0550] 子通道号

[0551] 子通道号是用于寻址子通道的系统唯一的 16 位值。该值在通道子系统的子通道组内是唯一的。子通道被 8 个 I/O 指令寻址:CANCEL SUBCHANNEL(取消子通道)、CLEAR SUBCHANNEL(清除子通道)、HALT SUBCHANNEL(暂停子通道)、MODIFY SUBCHANNEL(修改子通

道)、RESUME SUBCHANNEL(恢复子通道)、START SUBCHANNEL(开始子通道)、STORE SUBCHANNEL(存储子通道)和 TEST SUBCHANNEL (测试子通道)。相对于特定 I/O 设备的所有 I/O 功能由程序通过指定被分配给 I/O 设备的子通道来指定。每个子通道组中的子通道总是被分配单个范围的连续编号内的子通道号。最低编号的子通道是子通道 0。通道子系统的最高编号的子通道具有等于比提供的子通道的数量少 1 的子通道编号。在每个子通道组中最多可提供 65536 个子通道。通常,子通道编号仅被用于 CPU 程序和通道子系统之间的通信。

[0552] 设备号

[0553] 具有分配给其的 I/O 设备的每个子通道也包含被称为设备号的参数。设备号是 16 位的值,其在设备被分配给子通道时被分配作为子通道的参数中的一个。设备号唯一地向程序标识设备。

[0554] 设备号提供了识别设备的手段,这跟系统模型、配置或通道路径协议施加的任何限制无关。设备号被用在关于设备的在系统和系统操作者之间发生的通信中。例如,设备号由系统操作者输入,以指定将被用于初始程序加载的输入设备。

[0555] 编程说明 :设备号在设备安装时间被分配,且可具有任何值。但是,用户必须遵守对设备号分配的任何限制,其可被控制程序、支持程序或特定的控制单元或 I/O 设备所要求。

[0556] 设备标识符

[0557] 设备标识符是由通道子系统使用以与 I/O 设备通信的地址,其对于程序来说不明显(apparent)。所使用的设备标识符的类型取决于所提供的特定通道路径类型和协议。每个子通道包含一个或多个设备标识符。

[0558] 对于并行 I/O 接口类型的通道路径,设备标识符被称为设备地址且由一个 8 位值组成。对于 ESCON I/O 接口,设备标识符由 4 位控制单元地址和 8 位设备地址组成。对于 FICON I/O 接口,设备标识符由 8 位控制单元映像 ID 和 8 位设备地址组成。对于 FICON 转换的 I/O 接口,设备标识符由 4 位控制单元地址和 8 位设备地址组成。

[0559] 设备地址识别与子通道关联的特定 I/O 设备(且在并行 I/O 接口上是控制单元)。设备地址可识别例如特定磁带驱动器、盘访问装置或传输线。范围 0-255 中的任何编号可被分配为设备地址。

[0560] 光纤通道扩展

[0561] 光纤通道扩展(FCX)设施是可选的设施,其提供由传输控制字(TCW)组成的通道程序的形成,所述传输控制字指定传输命令控制块(TCCB)和传输状态块(TSB)。TCCB 包括传输命令区域(TCA),其包含多达 30 个形式为设备命令字(DCW)的 I/O 命令的列表。TCW 和其 TCCB 可指定读或写操作。除了 IRB, TSB 包含完成状态和与 TCW 通道程序关联的其他信息。

[0562] FCX 设施提供直接或间接指定 TCCB、输入数据存储区域和输出数据存储区域中的任何一个或全部的能力。当存储区域被直接指定时,TCW 指定单个、连续存储块的位置。当存储区域被间接指定时,TCW 指定一个或多个传输间接数据地址字(TIDAW)的列表的位置。TIDAW 列表和由列表中每个 TIDAW 指定的存储区域被限制跨过 4K 字节的边界。

[0563] FCX 设施也提供询问操作,其可被取消子通道(CANCEL SUBCHANNEL)指令启动以确定 I/O 操作的状态。

[0564] I/O 命令字

[0565] I/O 命令字指定命令并包含与命令相关的信息。当 FCX 设施已被安装时, 存在两种基本形式的 I/O 命令字, 其是通道命令字(CCW) 和设备命令字(DCW)。

[0566] CCW 是 8 字节的长度, 并指定将被执行的命令。对于启动特定操作的命令, CCW 也指定与操作相关的存储区域、数据字节的计数、当命令完成时将采取的动作、以及其他选项。所有的 I/O 设备识别 CCW。

[0567] DCW 长度是 8 字节, 并指定将被执行的命令、数据字节的计数和其他选项。支持 FCX 的 I/O 设备识别 DCW。

[0568] 传输命令字(TCW)

[0569] TCW 指定传输命令控制块(TCCB), 其包含将被传输给 I/O 设备并由其执行的命令列表。TCW 也指定用于 TCCB 中的命令的存储区域, 以及包含 I/O 操作的状态的传输状态块(TSB)。

[0570] 通道程序组织

[0571] 当 FCX 设施未被安装时, 有单个形式的通道程序, 其是 CCW 通道程序。当 FCX 设施已被安装时, 存在额外形式的通道程序, 其是 TCW 通道程序。两种形式的通道程序都在以下描述。

[0572] CCW 通道程序

[0573] 由一个或多个 CCW 组成的通道程序被称为 CCW 通道程序(CCP)。这样的通道程序包含一个或多个 CCW, 其被逻辑连接并被安排为由通道子系统顺序执行。

[0574] TCW 通道程序

[0575] 由单个 TCW 组成的通道程序被称为 TCW 通道程序。TCW 指定传输命令控制块(TCCB), 其包含 1 到 30 个 DCW。TCCB 内的 DCW 被逻辑连接, 且被安排为顺序执行。对于指定控制信息的 DCW, TCCB 也包含用于这些命令的控制信息。TCW 也指定用于 DCW 的一个或多个存储区域, 所述 DCW 指定从设备或到设备的数据传送, 以及用于完成状态的传输状态块(TSB) 的位置。TCCB 和用于数据传送的存储区域可被指定为连续或不连续的存储。

[0576] TCW 也指定用于完成状态的 TSB。

[0577] I/O 操作的执行

[0578] I/O 操作被启动, 且被具有 4 种类型的格式的信息控制: 指令 START SUBCHANNEL(开始子通道)、传输命令字、I/O 命令字以及顺序。START SUBCHANNEL 指令由 CPU 执行, 且是 CPU 程序的一部分, 该 CPU 程序监督来自管理或处理 I/O 数据的其他程序的、用于 I/O 操作的请求流。

[0579] 当 START SUBCHANNEL 被执行时, 参数被传递给目标子通道, 请求通道子系统利用与子通道关联的 I/O 设备执行开始功能。通道子系统通过使用子通道处的信息执行开始功能, 所述信息包括在 START SUBCHANNEL 指令的执行期间传递的信息, 以找到到设备的可访问的通道路径。一旦设备已被选择, 对于 CCW 通道程序, I/O 操作的执行通过通道子系统和 I/O 设备对 CCW 的解码和执行而完成, 或者对于 TCW 通道程序, 通过由通道子系统传输 TCCB 到 I/O 设备, 并由设备解码并执行 DCW 而完成。I/O 命令字以及传输命令字从主存储器中获取, 尽管 CCWDCW 的命令代码中的修改符位可为设备处的操作的执行指定取决于设备的条件。

[0580] 特定于设备的操作,诸如重绕磁带或将访问机构放置在磁盘驱动上,由 I/O 设备解码和执行的命令指定。命令可作为控制命令的命令代码中的修改符位被传递到设备中,作为在控制或写操作期间的数据被传送给设备,或可通过其他手段对设备可用。

[0581] 开始功能启动

[0582] CPU 程序利用指令 START SUBCHANNEL 启动 I/O 操作。该指令将操作请求块(ORB)的内容传递给子通道。如果 ORB 指定 CCW 通道程序,ORB 的内容包括子通道键、将被执行的第一个 CCW 的地址和 CCW 的格式的指定。CCW 指定将被执行的命令和将被使用的存储区域(如有)。如果 ORB 指定 TCW 通道程序,ORB 的内容包括子通道键和将被执行的 TCW 的地址。TCW 指定 TCCB,其包含将被传输给设备来执行的命令、将用于数据传送的一个或多个存储区域(如有),以及包含 I/O 操作的状态的 TSB。

[0583] 当 ORB 内容已被传递给子通道,START SUBCHANNEL 的执行完成。该指令执行的结果由程序状态字中设置的条件代码指示。

[0584] 当设施变得可用且 ORB 指定 CCW 通道程序,通道子系统获取第一个 CCW 并根据 ORB 中指定的格式位对其进行解码。如果格式位是零,指定格式 0CCW。如果格式位是 1,指定格式 1CCW。格式 0 和格式 1CCW 包含相同的信息,但是字段在格式 1CCW 中被不同地安排,这样 31 位地址可在 CCW 中被直接指定。当设施变得可用且 ORB 指定 TCW 通道程序,通道子系统获取指定的 TCW 并将指定的 TCCB 传输给设备。由 TCW 指定用于传送数据到设备或从设备传送数据的存储区域是 64 位地址。

[0585] 子通道操作模式

[0586] 存在两种子通道操作模式。当 FCX 设施被安装,且作为指定了 TCW 通道程序的 START SUBCHANNEL 指令的执行结果,开始功能在子通道处被设置,子通道进入传输模式。子通道保持在传输模式,直到开始功能在子通道处被重置。在所有其他时刻,子通道处于命令模式。

[0587] 路径管理

[0588] 如果 ORB 指定 CCW 通道程序,且第一个 CCW 传递特定的有效性测试且不具有被指定为 1 的暂停标志,或者如果 ORB 指定 TCW 通道程序,且指定的 TCW 传递特定的有效性测试,通道子系统通过从可供选择的通道路径组选择通道路径,来尝试设备选择。识别设备标识符的控制单元将其自己逻辑连接到通道路径并响应于其选择。

[0589] 如果 ORB 指定 CCW 通道程序,通道子系统通过通道路径发送 CCW 的命令代码部分,且设备以指示命令是否可被执行的状态字节进行响应。此时控制单元可逻辑地从通道路径断开,或它可保持连接以启动数据传送。

[0590] 如果 ORB 指定 TCW 通道程序,通道子系统使用指定的 TCW 中的信息以将 TCCB 传递到控制单元。TCCB 的内容被通道子系统忽略,且仅对控制单元和 I/O 设备有意义。

[0591] 如果由于忙指示或通道不可操作条件的结果,没有发生尝试的选择,通道子系统试图通过可替换的通道路径(如果其可用)选择设备。当选择已在所有可供选择的路径上被尝试且忙条件持续,操作保持为待决直到通道空闲。如果通道不可操作条件在尝试了设备选择的一条或多条路径上被检测,程序被随后的 I/O 中断警告。I/O 中断发生在通道程序的执行时(假设设备在可选的通道路径上被选择)或作为因为路径不可操作条件在尝试了设备选择的所有通道路径上被检测执行被放弃的结果。

[0592] 通道程序执行

[0593] 如果命令在设备上被启动且命令执行不要求任何数据被传送到设备或从设备被传送,设备可在收到命令代码后立即信号通知操作结束。在涉及数据传送的操作中,子通道被建立,这样通道子系统将响应于来自设备的服务请求并采取对操作的进一步控制。

[0594] I/O 操作可涉及到或来自由单个 CCW 或 TCW 指定的一个存储区域的数据传送,或者到或来自若干不连续的存储区域的数据传送。在后一种情况中,通常 CCW 列表被用于 I/O 操作的执行,每个 CCW 指定连续的存储区域且 CCW 被数据链接(chaining)耦合。数据链接被 CCW 中的标志指定,且使得通道子系统一旦当前 CCW 指定的存储区域被耗尽或填充,获取另一个 CCW。由在数据链接上获取的 CCW 指定的存储区域属于 I/O 设备处已在进行中的 I/O 操作,且当新的 CCW 被获取时,I/O 设备不会被通知。

[0595] 在 CCW 格式中提供了规定,以使程序员可指定当 CCW 被解码时,通道子系统尽快请求 I/O 中断,由此向 CPU 程序通知链接已至少前进了跟 CCW 在通道程序前进的一样远。

[0596] 为了补充 CPU 中的动态地址转换,CCW 间接数据寻址和修改的 CCW 间接数据寻址被提供。

[0597] 当 ORB 指示 CCW 通道程序以及 CCW 间接数据寻址被使用时,CCW 中的标志指定间接数据地址列表被用于指定用于该 CCW 的存储区域。每次到达存储块的边界时,列表被引用以确定将被使用的下一个存储块。ORB 指定每个存储块的大小是 2K 字节还是 4K 字节。

[0598] 当 ORB 指定 CCW 通道程序且修改的 CCW 间接数据寻址被使用时,ORB 中的标志和 CCW 中的标志指示修改的间接数据地址列表将被用来指定用于该 CCW 的存储区域。每次达到指定用于存储块的字节计数时,引用该列表以确定下一个将被使用的存储块。与当间接数据寻址被使用不同,块可在任何边界上被指定,且长度可多达 4K,假设跨过 4K 字节边界的数据传送未被指定的话。

[0599] 当 ORB 指定 TCW 通道程序且传输间接数据寻址被使用,TCW 中的标志指示传输间接数据地址列表是否被使用以指定包含 TCCB 的存储区域,以及传输间接数据地址列表是否被使用以指定与 TCCB 中的 DCW 关联的数据存储区域。每次达到指定用于存储块的字节的计数时,对应的传输间接数据地址列表被引用以确定将被使用的下一个存储块。

[0600] CCW 间接数据寻址和修改的 CCW 间接数据寻址允许基本上相同的 CCW 序列被用于利用在 CPU 中活动的动态地址转换运行的程序,就像其在 CPU 利用等效的连续真实存储来运行的情况下将被使用那样。CCW 间接数据寻址允许程序指定数据块具有多达 264-1 的绝对存储地址,而不考虑是否在 ORB 中指定了格式 0 或格式 1CCW。修改的 CCW 间接数据寻址允许程序指定数据块具有多达 264-1 的绝对存储地址,而不考虑是否在 ORB 中指定了格式 0 或格式 1CCW。

[0601] 一般而言,I/O 操作或操作链的执行涉及多达 3 级的参与:

[0602] 1. 除了由于 CPU 和通道子系统的集成引起的效果,CPU 在 START SUBCHANNEL 的执行期间是忙的,其持续直到寻址的子通道已被传递 ORB 内容。

[0603] 2. 对于新的 START SUBCHANNEL,从接收到 ORB 内容直到初级中断条件在子通道处被清除,子通道是忙的。

[0604] 3. 从设备处启动第一个操作直到子通道变得暂停或次级中断条件被放置在子通道处,I/O 设备是忙的。在暂停子通道的情况下,当恢复执行暂停的通道程序时,设备再次

变得忙。

[0605] I/O 操作的结束

[0606] I/O 操作的结束通常由两种状态条件指示：通道端和设备端。通道端条件指示 I/O 设备已接收或提供与操作相关的所有数据，且不再需要通道子系统设施。该条件被称为初级中断条件，且在该情况下通道端是初级状态。通常，初级中断条件是涉及 I/O 操作且信号通知 I/O 操作或 I/O 操作链的子通道处处的结束的任何中断条件。

[0607] 设备端信号指示 I/O 设备已结束执行，且准备好执行另一操作。该条件被称为次级中断条件，且该情况下的设备端处于次级状态。通常次级中断条件是涉及 I/O 操作且信号通知 I/O 操作或操作链的设备处的结束的任何中断条件。次级中断条件可与初级中断条件并行发生或晚于其发生。

[0608] 与初始或次级中断条件并行地，通道子系统和 / 或 I/O 设备都可提供不平常情况的指示。

[0609] 信号通知 I/O 操作的结束的条件可由 I/O 中断引起程序的注意，或者当 CPU 被禁止用于 I/O 中断时，通过通道子系统的编程询问来引起程序的注意。在前一种情况下，这些条件造成存储 I/O 中断代码，其包含关于中断源的信息。在后一种情况下，中断代码被存储为测试待决中断 (TEST PENDING INTERRUPTION) 的执行结果。

[0610] 当识别初级中断条件时，通道子系统试图通过中断请求的方式向程序通知子通道包含描述子通道处的 I/O 操作的结束的信息。对于命令模式中断，该信息标识使用的最后一个 CCW，且可提供其剩余字节计数，由此描述所使用的主存储器的区段。对于传输模式中断，该信息标识当前 TCW 以及与通道程序关联的 TSB，其包含关于 I/O 操作的额外状态，诸如剩余字节计数。除了关于通道程序的信息，通道子系统和 I/O 设备都可提供作为初始或次级中断条件的一部分的不平常情况的额外指示。包含在子通道处的信息可被测试子通道 (TEST SUBCHANNEL) 的执行或存储子通道 (STORE SUBCHANNEL) 的执行存储。当被存储时，该信息被称为子通道状态字 (SCSW)。

[0611] 当使用 CCW 通道程序时的链接

[0612] 当 ORB 指定 CCW 通道程序时，设施被提供，以使程序利用单个 START SUBCHANNEL 指令启动 I/O 操作链的执行。当当前 CCW 指定命令链接且在执行期间没有检测到不平常条件时，接收到设备端信号使得通道子系统获取新的 CCW。如果 CCW 通过特定的有效性测试且暂停标签没有在新的 CCW 中被指定为 1，新命令的执行在设备处被启动。如果 CCW 未能通过有效性测试，新命令不被启动，命令链被暂停，且与新 CCW 关联的状态引起中断条件的产生。如果暂停标志被指定为 1 且该值由于暂停控制 (相关 ORB 的字 1 的位 4) 中的 1 值而是有效的，新命令的执行不会被启动，且命令链被结束。

[0613] 新命令的执行由通道子系统以与之前操作相同的方式被启动。使得在指定命令链的 CCW 引起的操作的结束时发生的结束信号不可用于程序，当另一个 I/O 操作被命令链启动时，通道子系统继续通道程序的执行。但是如果已检测到不平常的条件，抑制命令链，终止通道程序，生成中断条件，且使得引起终止的结束信号对程序可用。

[0614] 暂停和恢复功能向程序提供对于通道程序的执行的控制。暂停功能的启动由 ORB 中的暂停控制位的设置控制。当 ORB 中的暂停控制位是 1 且第一个 CCW 中或命令链期间获取的 CCW 中的暂停标志是 1 时，暂停功能在通道程序执行期间被信号通知给通道子系统。

[0615] 当通道子系统获取具有被有效地指定为 1(由于 ORB 中的暂停控制位的 1 值)的暂停标志的 CCW 时,发生暂停。该 CCW 中的命令不被发给 I/O 设备,且设备被信号通知命令链被结束。随后的恢复子通道(RESUME SUBCHANNEL)指令向通道子系统通知产生暂停的 CCW 可能已被修改,且该通道子系统必须重新获取 CCW 并检查暂停标志的当前设置。如果在该 CCW 中暂停标志被发现是零,通道子系统恢复命令链针对 I/O 设备的执行。

[0616] 使用 TCW 通道程序时的链接

[0617] 当 ORB 指定 TCW 通道程序时,设施也被提供以便程序利用单个 START SUBCHANNEL 指令启动设备操作链的执行。命令链接可被指定用于由单个 TCW 指定的那些 DCW。当当前 DCW 指定命令链且在操作期间没有检测到不平常条件时,识别 DCW 的成功执行使得当前 TCCB 中的下一个 DCW 被处理。

[0618] 如果下一个 DCW 通过特定的有效性测试,新命令的执行在设备处启动,且 DCW 变成当前 DCW。如果 DCW 未能通过有效性测试,新命令不被启动,命令链接被抑制,通道程序被终止,且与新 DCW 关联的状态引起中断条件被生成。

[0619] 新命令的执行以与之前的操作相同的方式被启动。使得发生在由不是最后指定的 DCW 的 DCW 引起的操作的结束处的结束信号不对程序可用。当另一个 I/O 操作被命令链启动时,通道程序的执行继续。但是如果检测到不平常条件时,命令链被抑制,通道程序被终止,中断条件被生成,且使得状态对识别不平常条件的程序可用。

[0620] I/O 操作的提前结束

[0621] 通道程序执行可被 CANCEL SUBCHANNEL、HALT SUBCHANNEL 或 CLEAR SUBCHANNEL 提前终止。如果通道程序没有在设备处被启动,CANCEL SUBCHANNEL 的执行使得通道子系统终止在子通道处的开始功能。当开始功能被 CANCEL SUBCHANNEL 的执行终止时,响应于 CANCEL SUBCHANNEL 指令,通道子系统设置条件代码 0。HALT SUBCHANNEL 的执行使得通道子系统发出暂停信号给 I/O 设备,并终止子通道处的通道程序执行。当通道程序执行被 HALT SUBCHANNEL 的执行终止时,通过 I/O 中断请求的方式,程序被通知终止。当子通道处于命令模式时,当设备呈现用于中断的操作的状态时,中断请求生成。当子通道处于传输模式时,中断请求被立即生成。但是如果在命令结链期间在设备端接收到但在下一个命令被传送给设备之前,暂停信号被发出给设备,在设备被信号通知后生成中断请求。在后一种情况中,SCSW 的设备状态字段将包含零。CLEAR SUBCHANNEL 的执行从子通道清除执行中的通道程序的指示,使得通道子系统发出清除信号到 I/O 设备,并使得通道子系统生成 I/O 中断请求以向程序通知清除功能的完成。

[0622] I/O 中断

[0623] 引起 I/O 中断请求的条件对于 CPU 中的活动是异步的,且多于一个的条件可同时发生。条件被保留在子通道,直到被 TEST SUBCHANNEL 或 CLEAR SUBCHANNEL 清除,或被 I/O 系统重置。

[0624] 当 I/O 中断条件已被通道子系统识别并在子通道被指示,使得 I/O 中断请求变得对于在子通道处指定的 I/O 中断子通道是待决的。使得中断对其变得待决的 I/O 中断子类通过使用 MODIFY SUBCHANNEL 而被编程控制。待决的 I/O 中断可被任何针对来自其 I/O 中断子类的中断被使能的 CPU 接受。每个 CPU 在控制寄存器 6 中具有 8 个掩码位,其控制该 CPU 针对 8 个 I/O 中断子类中的每个的使能,其中,I/O 掩码,即 PSW 中的位 6,是用于 CPU 的

主 I/O 中断掩码。

[0625] 当 I/O 中断发生在 CPU 时, I/O 中断代码被存储在该 CPU 的 I/O 通信区域, 且 I/O 中断请求被清除。I/O 中断代码识别子通道, 对于该子通道中断是待决的。引起中断请求产生的条件可随后被 TEST SUBCHANNEL 或 STORE SUBCHANNEL 显式地从子通道获取。

[0626] 当对应地 I/O 中断子类被使能但 PSW 使得 I/O 中断被禁止时, 待决的 I/O 中断请求也可被 TEST PENDING INTERRUPTION 清除, 或当 CPU 针对来自对应的 I/O 中断子类的 I/O 中断被禁止时, 该请求也可被 TEST SUBCHANNEL 禁止。待决的 I/O 中断请求也可被 CLEAR SUBCHANNEL 清除。CLEAR SUBCHANNEL 和 TEST SUBCHANNEL 两者也清除子通道处的保留的中断条件。

[0627] 通常, 除非中断请求被 CLEAR SUBCHANNEL 清除, 程序发出 TEST SUBCHANNEL 以获取关于操作执行的信息。

[0628] 清除子通道

[0629] 指定的子通道被清除, 当前开始或停止功能(如有)在指定的子通道处被终止, 且通道子系统被信号通知以在指定的子通道和在相关设备处异步地执行清除功能。

[0630] 通用寄存器 1 包含子系统识别字(SID), 其指定将被清除的子通道。

[0631] 如果开始或暂停功能正在进行, 其在子通道处终止。

[0632] 使得子通道不再状态待决。所有的活动, 如 SCSW 的活动控制字段所指示的, 在子通道处被清除, 除非使得子通道变得清除待决。进行中的任何功能, 如在 SCSW 的功能控制字段中指示的, 在子通道处被清除, 除了由于该指令的执行而将被执行的清除功能。

[0633] 当子通道在传输模式中运行且条件代码 2 被设置, CPU 可信号通知通道子系统异步地执行询问功能, 并结束指令。

[0634] 通道子系统被信号通知异步地执行清除功能。清除功能在以下“相关功能”小节中被概述, 并在此后被详细描述。

[0635] 条件代码 0 被设置以指示上述动作已被采取。

[0636] 相关功能

[0637] 在 CLEAR SUBCHANNEL 执行之后, 通道子系统异步地执行清除功能。如果条件允许, 通道子系统选择通道路径, 并试图发出清除信号给设备以终止 I/O 操作(如有)。子通道随后变为状态待决。通道子系统遇到的排除发出清除信号到设备的条件不会阻止子通道变得状态待决。

[0638] 当由于执行清除功能, 子通道变为状态待决时, 与相关设备的数据传送(如有)已被终止。当产生的状态被 TEST SUBCHANNEL 清除时存储的 SCSW 具有存储为 1 的清除功能位。如果通道子系统可确定清除信号被发出给设备, 清除待决位作为零被存储在 SCSW 中。否则, 清除待决位被存储为 1, 且其他指示被提供, 其描述所遇到的条件的进一步细节。

[0639] 对于被 CLEAR SUBCHANNEL 终止的开始功能, 没有积聚测量数据, 且设备连接时间没有被存储在用于子通道的扩展状态字中。

[0640] 特殊条件

[0641] 当子通道对于 CLEAR SUBCHANNEL 不可操作时, 条件代码 3 被设置, 且不采取其他动作。当子通道没有在通道子系统中被提供、没有有效设备号被分配给它、或未被使能时, 子通道对于 CLEAR SUBCHANNEL 不可操作用。

[0642] CLEAR SUBCHANNEL 可遇到下述或下列程序异常。当多子通道组设施未被安装时，通用寄存器 1 的位 32-47 必须包含 0001hex；否则，操作数异常被识别。

[0643] 当多子通道组设施已被安装时，通用寄存器 1 的位 32-44 必须包含零，位 45-46 必须包含有效值，且位 47 必须包含值 1；否则，操作数异常被识别。

[0644] 产生的条件代码：

[0645] 0 功能被启动

[0646] 1 —

[0647] 2 —

[0648] 3 不可操作

[0649] 程序异常：

[0650] • 操作数

[0651] • 特许操作

[0652] 清除功能

[0653] 随着 CLEAR SUBCHANNEL 的执行，通道子系统执行清除功能。清除功能的执行包括（1）执行路径管理操作，（2）修改子通道处的字段，（3）发出清除信号给相关设备，以及（4）使得子通道变为状态待决，指示清除功能的完成。

[0654] 清除功能路径管理

[0655] 路径管理操作作为清除功能的一部分被执行，以检查用于相关子通道的通道路径条件，并试图选择可用的通道路径，在其上清除信号可被发出给相关设备。

[0656] 通道路径条件按以下顺序被检查：

[0657] 1. 如果通道子系统活动地与将被信号通知的设备通信或尝试建立与其的活动通信，选择使用中的通道路径。

[0658] 2. 如果通道子系统正在接受来自将被信号通知的设备的不再忙指示（其将不会引起中断条件被识别）的过程中，且相关的子通道对于任何通道路径没有忠诚度（allegiance），选择使用中的通道路径。

[0659] 3. 如果相关的子通道对于一通道路径具有专用忠诚度，该通道路径被选择。

[0660] 4. 如果相关的子通道对于一条或多条通道路径具有工作忠诚度，选择这些通道路径中的一条。

[0661] 5. 如果相关子通道对于任何通道路径没有忠诚度，如果最后使用的通道路径被指示，且如果该通道路径可供选择，该通道路径被选择。如果该通道路径不可供选择，则没有通道路径被选择，或通道路径从可供选择的一组通道路径（如有）中被选择（尽管没有指示最后使用的通道路径）。

[0662] 6. 如果相关子通道对于任何通道路径没有忠诚度，如果没有指示最后使用的通道路径，且如果存在可供选择的一条或多条通道路径，选择这些通道路径中的一条。

[0663] 如果以上列出的通道路径条件都不试用，不选择通道路径。对于项目 4，对于位于指定条件下的项目 5，且对于项目 6，通道子系统从一组通道路径中选择通道路径。在这些情况下，如果以下条件不适用，通道子系统可尝试选择通道路径：

[0664] 1. 通道路径终止条件对于通道路径存在。

[0665] 2. 对于并行或 ESCON 通道路径：另一条子通道对于通道路径具有有效忠诚度。

[0666] 对于 FICON 通道路径 : 通道路径当前被使用以与最大数量的可具有并行活动通信的子通道通信。

[0667] 3. 将被信号通知的设备被附加到类型 1 控制单元, 且用于被附加到相同的控制单元的另一个设备的子通道对于相同的通道路径具有忠诚度, 除非该忠诚度是工作忠诚度, 且初级状态已被该子通道接受。

[0668] 4. 将被信号通知的设备被附加到类型 3 控制单元, 且用于被附加到相同的控制单元的另一个设备的子通道对于相同的通道路径具有专用忠诚度。

[0669] 清除功能子通道修改

[0670] 在清除功能执行期间, 子通道处的路径管理控制指示被修改。有效地, 该修改在尝试选择通道路径后但是在尝试选择设备以发出清除信号前发生。被修改的路径管理控制指示如下 :

[0671] 1. 子通道处的所有 8 个可能的通道路径的状态被设置为对于该子通道是可操作的。

[0672] 2. 最后使用的路径指示被重置以指示没有最后使用的通道路径。

[0673] 3. 路径不可操作条件(如有)被重置。

[0674] 清除功能信令和完成

[0675] 在尝试选择通道路径和路径管理控制字段的修改后, 如果条件允许, 通道子系统尝试选择设备以发出清除信号。与子通道和选择的通道路径(如有)关联的条件影响(1)是否做出尝试以发出清除信号, 以及(2)尝试发出清除信号是否成功。不考虑这些条件, 子通道随后被设置状态待决, 且清除功能的执行完成。这些条件及其对于清除功能的效果被如下描述 :

[0676] 不尝试发出清除信号 : 如果存在任何以下条件, 通道子系统不试图发出清除信号给设备 :

[0677] 1. 未选择通道路径。

[0678] 2. 选择的通道路径不再可供选择。

[0679] 3. 对于选定的通道路径存在通道路径终止条件。

[0680] 4. 对于并行和 ESCON 通道路径 : 选定的通道路径当前被使用以活动地与不同设备通信。对于 FICON 通道路径 : 选定的通道路径当前被使用以活动地与最大数量的可具有并行活动通信的设备通信。

[0681] 5. 将被信号通知的设备被附加到类型 1 控制单元, 且用于被附加到相同的控制单元的另一个设备的子通道对于相同的通道路径具有忠诚度, 除非该忠诚度是工作忠诚度且初级状态已被该子通道接受。

[0682] 6. 将被信号通知的设备被附加到类型 3 控制单元, 且用于被附加到相同的控制单元的另一个设备的子通道对于相同的路径具有专用忠诚度。

[0683] 如果存在上述任何条件, 子通道保持清除待决且被设为状态待决, 且清除功能的执行完成。

[0684] 尝试发出清除信号没有成功 : 当通道子系统试图发出清除信号给设备时, 由于以下条件, 尝试可能不成功 :

[0685] 1. 当通道子系统试图选择设备以发出清除信号时, 控制单元或设备提示忙条件。

[0686] 2. 当通道子系统试图选择设备以发出清除信号时,路径不可操作条件被识别。

[0687] 3. 当通道子系统试图发出清除信号时,遇到错误条件。

[0688] 如果任何上述条件存在,且通道子系统确定试图发出清除信号不成功,或不能确定尝试是否成功,子通道保持清除待决,且被设为状态待决,且清除功能的执行完成。

[0689] 尝试发出清除信号是成功的:当通道子系统确定尝试发出清除信号是成功的,子通道不再是清除待决的,且被设为状态待决,且清除功能的执行完成。当子通道变为状态待决时,针对相关设备的 I/O 操作(如有)已被终止。

[0690] 编程说明:在清除功能执行后,被设备呈现给通道子系统的任何非零状态,除了仅控制单元端之外,作为未被请求的告警状态被传送给程序。包含仅控制单元端或零状态的未被请求的状态不会被呈现给程序。

[0691] 修改子通道

[0692] 包含在子通道信息块(SCHIB)中的信息被放置在子通道处的程序可修改字段。结果,对于该子通道,程序影响相对于清除、暂停、恢复和开始功能和特定 I/O 支持功能的 I/O 处理的特定方面。

[0693] 通用寄存器 1 包含子系统识别字(SID),其指定将被修改的子通道,如被 SCHIB 的特定字段指定的。第二操作数地址是 SCHIB 的逻辑地址,且必须被指定在字边界上;否则,识别规格异常。

[0694] 由于 SCHIB 信息在子通道中的放置而可被影响的通道子系统操作是:

[0695] • I/O 处理(E 字段)

[0696] • 中断处理(中断参数和 ISC 字段)

[0697] • 路径管理(D、LPM 和 POM 字段)

[0698] • 监视和地址界限检验(测量 - 块索引、LM 和 MM 字段)

[0699] • 测量块格式控制(F 字段)

[0700] • 扩展测量字模式使能(X 字段)

[0701] • 并行感测设施(S 字段)

[0702] • 测量块地址(MBA)

[0703] 字 1 的位 0、6 和 7,以及 SCHIB 操作数的字 6 的位 0-28 必须是零,且字 1 的位 9 和 10 必须不同时为 1。当扩展 I/O 测量块设施被安装,且格式 1 测量块被指定时,字 11 的位 26-31 必须被指定为零。当扩展 I/O 测量块设施未被安装时,字 6 的位 29 必须被指定为零;否则,识别操作数异常。当扩展的 I/O 测量字设施未被安装时,或被安装但未被使能时,字 6 的位 30 必须被指定为零;否则,操作数异常被识别。SCHIB 的剩余字段被忽略,且不影响 MODIFY SUBCHANNEL 的处理。

[0704] 设置条件代码 0 以指示来自 SCHIB 的信息已被放置在子通道处的程序可修改字段中,除非当指定子通道处的设备号有效位(V)是零,则条件代码 0 被设置,且来自 SCHIB 的信息不被放置在程序可修改字段中。

[0705] 特殊条件

[0706] 当子通道是状态待决时,条件代码 1 被设置,且不采取其他动作。

[0707] 当清除、暂停或开始功能在子通道处进行时,条件代码 2 被设置,且不采取其他动作。

[0708] 当子通道对于 MODIFY SUBCHANNEL 不可操作时, 条件代码 3 被设置且不采取其他动作。当子通道在通道子系统中没有被提供时, 子通道对于 MODIFY SUBCHANNEL 不可操作。

[0709] MODIFY SUBCHANNEL 可遇到如下所述或所列程序异常。

[0710] 在 SCHIB 的字 1 中, 位 0、1、6 和 7 必须是零, 且当地址界限检验设施被安装时, 位 9 和 10 必须不都为零。在 SHICB 的字 6 中, 位 0-28 必须是零。否则, 识别操作数异常。

[0711] 当扩展的 I/O 测量块设施被安装且格式 1 测量块被指定时, 字 11 的位 26-31 必须被指定为零; 否则, 识别操作数异常。当扩展的 I/O 测量块设施未被安装时, 字 6 的位 29 必须被指定为零; 否则, 操作数异常被识别。当扩展的 I/O 测量字设施未被安装时, 或被安装但未被使能时, 字 6 的位 30 必须被指定为零; 否则, 操作数异常被识别。

[0712] 当多子通道组设施未被安装时, 通用寄存器 1 的位 32-47 必须包含 0001hex; 否则, 操作数异常被识别。

[0713] 当多通道组设施被安装时, 通用寄存器 1 的位 32-44 必须包含零, 位 45-46 必须包含有效值, 且位 47 必须包含值 1; 否则, 操作数异常被识别。

[0714] 第二操作数必须在字边界上指定; 否则, 规格异常被识别。MODIFY SUBCHANNEL 的执行在所有寻址和保护异常上被抑制。

[0715] 产生的条件代码:

[0716] 0 功能完成

[0717] 1 状态待决

[0718] 2 忙

[0719] 3 不可操作

[0720] 程序异常:

[0721] • 访问(获取, 操作数 2)

[0722] • 操作数

[0723] • 特许操作

[0724] • 规格

[0725] 编程说明:

[0726] 1. 如果当相关子通道被禁止时设备信号通知 I/O 错误告警, 通道子系统发出清除信号给设备并丢弃 I/O 错误告警指示, 而不生成 I/O 中断条件。

[0727] 2. 如果当相关子通道被禁止时设备呈现未请求的状态, 该状态被通道子系统丢弃而不生成 I/O 中断条件。但是, 如果呈现的状态包含单元检验, 通道子系统发出用于相关子通道的清除信号, 且不生成 I/O 中断条件。当程序使用 MODIFY SUBCHANNEL 来使能子通道时, 这应被考虑。例如, 当子通道被禁止时存在的相关设备上的介质可能已被替换, 且因此程序应当验证该介质的完整性。

[0728] 3. 建议当 MODIFY SUBCHANNEL 设置条件代码 0 时, 程序通过随后发出 STORE SUBCHANNEL 来检验子通道的内容。使用 STORE SUBCHANNEL 是确定指定的子通道是否改变的方法。MODIFY SUBCHANNEL 设置条件代码 0 之后未能检验子通道可能导致程序不会预期发生的条件。

[0729] START SUBCHANNEL

[0730] 通道子系统被信号通知异步执行用于相关设备的开始功能, 且包含在指定的 ORB

中的执行参数被放置在指定的子通道。

[0731] 通用寄存器 1 包含子系统识别字, 其指定将被开始的子通道。第二操作数地址是 ORB 的逻辑地址, 且必须在字边界上指定; 否则, 规格异常被识别。

[0732] 包含在 ORB 中的执行参数被放置在子通道处。

[0733] 当 START SUBCHANNEL 被执行时, 子通道是状态待决, 仅具有次级状态, 且扩展状态字格式位(L)是零, 状态待决条件在子通道处被丢弃。

[0734] 使子通道变得开始待决, 且开始功能在子通道处被指示。如果第二操作数指定命令模式 ORB, 子通道保持在命令模式中。如果第二操作数指定传输模式 ORB, 子通道进入到传输模式。当子通道进入到传输模式时, 如果没有之前的专用忠诚度存在, LPUM 被设为零; 否则 LPUM 不被改变。

[0735] 逻辑地, 在设置条件代码 0 之前, 子通道处的路径不可操作条件(如有)被清除。

[0736] 通道子系统被信号通知异步地执行开始功能。开始功能在以下小节“相关功能”中被概念, 且在以后被详细描述。

[0737] 条件代码 0 被设置以指示已采取上述动作。

[0738] 相关功能

[0739] 在 START SUBCHANNEL 执行后, 通道子系统异步地执行开始功能。

[0740] 检验 ORB 内容的有效性, 除了必须包含全零的字段。在一些模型上, 必须包含零的 ORB 的字段被异步检验, 而不是在指令的执行期间。当无效字段被异步检测时, 子通道变为状态待决, 具有初级、次级和告警状态, 且延迟条件代码 1 和程序检验被指示。在这种情况下, I/O 操作或 I/O 操作链不在设备处被启动, 且当 SCSW 被 TEST SUBCHANNEL 的执行清除时, 条件被存储为 1 的开始待决位指示。

[0741] 在一些模型上, 异步地而不是在指令的执行期间测试路径可用性。当没有通道路径可供选择时, 子通道变为状态待决, 具有初级和次级状态, 且延迟条件代码 3 被指示。I/O 操作或 I/O 操作链不在设备处被启动, 且当 SCCW 被 TEST SUBCHANNEL 的执行清除时, 该条件被存储在为 1 的开始待决位指示。

[0742] 如果条件允许, 选择通道路径, 且启动在 ORB 中指定的通道程序的执行。

[0743] 特殊条件

[0744] 当 START SUBCHANNEL 被执行时且子通道是状态待决时, 条件代码 1 被设置, 且不采取其他动作。在一些模型上, 当子通道仅以次级状态是状态待决时, 条件代码 1 不被设置; 反而, 状态待决条件被丢弃。

[0745] 当开始、暂停或清除功能当前正在子通道处进行时, 条件代码 2 被设置, 且不采取其他动作。

[0746] 当子通道对于 START SUBCHANNEL 不可操作时, 条件代码 3 被设置, 且不采取其他动作。如果子通道在通道子系统中没有被提供, 没有与其相关的有效设备号或未被使能, 子通道对于 START SUBCHANNEL 不可操作。

[0747] 在一些模型上, 当没有通道路径可供选择时, 子通道对于 START SUBCHANNEL 也不可操作。在这些模型中, 缺少可用通道路径作为 START SUBCHANNEL 执行的一部分被检测。在其他模型上, 通道路径可用性仅作为异步开始功能的一部分被测试。

[0748] START SUBCHANNEL 可遇到以下所述或所列的程序异常。

[0749] 在命令模式 ORB 的字 1 中,位 26-30 必须是零,且在命令模式 ORB 的字 2 中,位 0 必须是零。否则,在一些模型上,操作数异常被识别。在其他模型上,I/O 中断条件被生成,指示程序检验,作为异步开始功能的一部分。

[0750] START SUBCHANNEL 也可遇到以下列出的程序异常。当多子通道组设施未被安装时,通用寄存器 1 的位 32-47 必须包含 0001hex;否则,操作数异常被识别。

[0751] 当多子通道组设施被安装时,通用寄存器 1 的位 32-44 必须包含零,位 45-46 必须包含有效值,且位 47 必须包含值 1;否则,操作数异常被识别。

[0752] 第二操作数必须在字边界上指定;否则,规格异常被识别。STARTSUBCHANNEL 的执行在所有寻址和保护异常上被抑制。

[0753] 产生的条件代码:

[0754] 0 功能启动

[0755] 1 状态待决

[0756] 2 忙

[0757] 3 不可操作

[0758] 程序异常

[0759] • 访问(获取,操作数 2)

[0760] • 操作数

[0761] • 特许操作

[0762] • 规格

[0763] 开始功能和恢复功能

[0764] 开始和恢复功能如下所述启动 I/O 操作。开始功能适用于在命令模式或传输模式中操作的子通道。恢复功能仅适用于在命令模式中操作的子通道。

[0765] 在执行 START SUBCHANNE 和 RESUME SUBCHANNEL 后,通道子系统分别执行开始和恢复功能,以针对相关设备启动 I/O 操作。开始或恢复功能的执行包括:(1)执行路径管理操作,(2)以相关设备执行 I/O 操作或 I/O 操作链,以及(3)使得子通道变为状态待决,指示开始功能的完成。开始功能启动在 ORB 中指定的通道程序的执行,ORB 转而被指定为 START SUBCHANNEL 的操作数,这与在引起暂停的 CCW 开始启动执行暂停的通道程序(如有)的恢复功能相反;否则,执行恢复功能就好象它是开始功能。

[0766] 开始功能和恢复功能路径管理

[0767] 在执行开始或恢复功能时由通道子系统来执行路径管理,以选择可被用于设备选择的可用通道路径,以使用该设备启动 I/O 操作。采取的动作如下:

[0768] 1. 如果子通道目前是开始待决且设备是活动的,开始功能在子通道处保持待决,直到从相关的设备接受用于之前的开始功能的次级状态并且使该子通道独自开始待决。当该状态被接受并且没有描述告警中断条件,不使子通道变为状态待决,并且随后启动执行待决开始功能。如果状态描述了告警中断条件,子通道变成具有次级和告警状态的状态待决,待决开始功能不会启动,延迟条件代码 1 被设置,并且开始待决位保持为 1。如果子通道当前独自开始待决,如下所述启动执行开始功能。

[0769] 2. 如果在子通道上存在对通道路径的专用忠诚度,通道子系统选择该路径以用于设备选择。如果在尝试选择设备时遇到忙条件并且在子通道上存在专用忠诚度,则开始功

能保持待决,直到针对该通道路径重置忙的内部指示。当忙的内部指示被重置,在该通道路径上启动执行待决的开始功能。

[0770] 3. 如果没有通道路径可供选择并且在子通道中不存在对通道路径的专用忠诚度,则通道路径不会被选择。

[0771] 4. 如果已经尝试了所有可供选择的通道路径,并且其中一个或多个被用于与其他设备的活动地通信,或者,如果通道子系统在这些通道路径的一个或多个上遇到控制单元忙或设备忙条件,或在这些通道路径的一个或多个上遇到这些条件的组合,开始功能在子通道上保持待决,直到通道路径、控制单元或设备在适当的情况下变得可用。

[0772] 5. 如果(1)在通道路径上启动开始功能,设备连接到类型 1 控制单元,并且(2)没有其他设备连接到相同的控制单元,该控制单元的子通道具有对相同通道路径的专用忠诚度或者对相同通道路径的工作忠诚度,其中,还未接收对于该子通道的初级状态,则选择该通道路径,如果它可供选择的话;否则,该通道路径不被选择。但是,如果到该设备的另一通道路径可供选择并且不存在如上所述的忠诚度,则通道路径被选择。如果没有其他通道路径可供选择,如果合适,开始和恢复功能保持待决,直到通道路径变得可用。

[0773] 6. 如果设备连接到类型 3 控制单元,并且如果至少一个其他设备连接到相同的控制单元,该控制单元的子通道具有对相同通道路径的专用忠诚度,则可以选择可供选择的另一通道路径,或者开始功能保持待决,直到用于该其他设备的专用忠诚度被清除。

[0774] 7. 如果选择了通道路径,并且在设备选择以启动执行待决的 CCW 通道程序的第一命令或者传输待决的 TCW 通道程序的 TCCB 期间接收到忙指示,则接收到忙指示的通道路径不会被再次用于该设备或控制单元(取决于接收到的设备忙或控制单元忙指示),直到忙的内部指示被重置。

[0775] 8. 如果在尝试选择设备以启动执行为开始指定或者 CCW 通道程序的恢复功能所隐含的第一命令或者启动针对 TCW 通道程序的开始功能的 TCCB 的传输的期间(如上动作 7 中所述),通道子系统接收到忙指示,它执行下列动作中的一个:

[0776] a. 如果设备被指定为在多路径模式下操作并且接收到的忙指示是设备忙,则开始或恢复功能保持待决,直到忙的内部指示被重置。

[0777] b. 如果设备被指定为在多路径模式下操作并且接收到忙指示是控制单元忙,或者如果设备被指定为在单路径模式下操作,则通道子系统通过选择可供选择的替代通道路径来尝试设备选择,并继续路径管理操作直到发起开始或恢复功能或者已经在可供选择的所有通道路径上尝试设备选择。如果在所有可供选择的通道路径都已被选择之后,通道子系统还未启动开始或恢复功能,则开始或恢复功能保持待决,直到忙的内部指示被重置。

[0778] c. 如果子通道具有专用忠诚度,则应用 15-20 页上的动作 2。

[0779] 9. 当在尝试选择以传送 CCW 通道程序的第一个命令或者传送 TCW 通道程序的 TCCB 的期间,设备看起来不可操作且相应的通道路径对于子通道可以操作时,路径不可操作条件被识别,并且通道路径的状态在子通道处从对于子通道可操作改变为对于子通道不可操作。子通道上的路径不可操作条件(如有)被保留,直到子通道接下来变为清除待决、开始待决或恢复待决(如果子通道被暂停),此时路径不可操作条件被清除。但是,如果相应的通道路径对于子通道不可操作,路径不可操作条件不被识别。当在尝试选择以在可供选择的通道路径上传送第一个命令或 TCCB 期间设备看起来不可操作时,出现下列动作中的一

个：

[0780] a. 如果对该通道路径存在专用忠诚度，则它是可供选择的唯一通道路径；因此，放弃启动开始或恢复功能的进一步尝试，且中断条件被识别。

[0781] b. 如果不存在专用忠诚度并且存在还未被尝试的可供选择的替代通道路径，则选择其中一个通道路径以尝试设备选择并传送 CCW 通道程序的第一个命令或 TCW 通道程序的 TCCB。

[0782] c. 如果不存在专用忠诚度、没有还未被尝试的可供选择的替代通道路径、并且设备看起来在已经尝试的至少一个通道路径上看来可操作，则开始和恢复功能在子通道上保持待决，直到通道路径、控制单元、或设备在适当的情况下变得可用。

[0783] d. 如果不存在专用忠诚度、没有还未被尝试的可供选择的替代通道路径、并且设备看起来在已经尝试的所有通道路径上看起来不可用，则放弃启动开始或恢复功能的进一步尝试，且中断条件被识别。

[0784] 10. 当子通道是活动的并且将针对设备启动 I/O 操作，如果在子通道上未指定多路径模式，则根据 LPUM 指示来进行所有设备选择。例如，如果为 CCW 通道程序指定了命令链接，则通道子系统在相同通道路径上传送描述了的 I/O 操作链的第一个和所有后续命令。

[0785] I/O 操作的执行

[0786] 在选择通道路径后，如果条件允许，通道子系统启动执行针对相关设备的 I/O 操作。额外的 I/O 操作的执行可以在第一次 I/O 操作的启动和执行之后。

[0787] 对于在命令模式下操作的子通道，通道子系统可以执行七种类型的命令：写入、读取、往回读、控制、感知、感知 ID 以及通道中的传送。除了通道中的传送，每个命令启动相应的 I/O 操作。除了通道程序执行在子通道上暂停的期间，从接受第一命令直到在子通道上识别初级中断条件，子通道都是活动的。如果在接受第一命令之前识别初级中断条件，子通道不会变成活动的。通常，初级中断条件是由通道结束信号或者在命令链接的情形下是由用于链中最后的 CCW 的通道结束信号引起的。设备是活动的，直到在子通道中识别次级中断条件。通常，次级中断条件是由设备结束信号或者在命令链接的情形下是由用于链中最后的 CCW 的设备结束信号引起的。对于在传输模式下操作的子通道，通道子系统可以传输六种类型的命令来执行：写入、读取、控制、感知、感知 ID 以及询问。每种命令启动相应的设备操作。当一个或多个命令在 TCCB 中被发送到 I/O 设备，子通道保持开始待决，直到出现初级状态。

[0788] 编程说明

[0789] 在单路径模式下，用于 I/O 操作或者 I/O 操作链的命令、数据和状态的所有传送发生在第一个命令在其上被传送到设备的通道路径上。

[0790] 当设备安装了动态重连特征，可以在多路径模式下执行 I/O 操作或 I/O 操作链。为了在多路径模式下操作，之前必须已经针对 SCHIB 的字 1 的位 13 被指定为 1 的子通道来执行 MODIFY SUBCHANNEL。此外，必须通过执行对于该类型的设备来说合适的特定的依赖于模式的命令，针对多路径模式来设置该设备。用于处理多路径模式操作的一般过程如下：

[0791] 1. 设置

[0792] a. 必须由设备在每个通道路径上成功地执行设置多路径模式类型的命令，该通道路径是被设置的多路径组的成员；否则，操作的多路径模式会在子通道上给出不可预测的

结果。如果因为任何原因,在多路径组中不包含到设备的一个或多个物理可用的通道路径,当子通道在多路径模式下操作时,这些通道路径必然不可供选择。可以通过在执行 MODIFY SUBCHANNEL 之前在 SCHIB 中或者在执行 START SUBCHANNEL 之前在 ORB 中将相应的 LPM 位设置为零,使通道路径不可供选择。

[0793] b. 当设置多路径模式类型的命令被传送到设备,必须只有单个通道路径逻辑可用,以避免替代的通道路径选择来执行该开始功能;否则,通道子系统可以在多于一个通道路径上检测到设备忙条件,这会导致后续多路径模式操作的不可预测的结果。每当多路径组的成员关系被改变,该类型的设置过程应被使用。

[0794] 2. 离开多路径模式

[0795] 为了离开多路径模式并且继续在单路径模式下处理,可以使用下列两个过程中的任一个:

[0796] a. 可以针对多路径组的任何通道路径来执行解散多路径模式类型的命令。该命令之后必须:(1)执行 MODIFY SUBCHANNEL,且 SCHIB 的字 1 的位 13 被指定为零,或者(2)仅将单通道路径指定为在 LPM 中逻辑可用。当设备在单路径模式下运行时,一定不能在具有可供选择的多条通道路径的多路径模式中操作的子通道上执行开始功能;否则,在子通道上会出现该功能或后续的开始功能的不可预测的结果。

[0797] b. 在多路径组的每个通道路径上执行放弃多路径模式类型的命令(与设置相反)。该命令之后必须:(1)执行 MODIFY SUBCHANNEL,且 SCHIB 的字 1 的位 13 被指定为零,或者(2)仅将单个通道路径指定为在 LPM 中逻辑可用。当设备在单路径模式下运行时,一定不能在具有可供选择的多条通道路径的多路径模式中操作的子通道上执行开始功能;否则,在子通道上会出现该功能或后续的开始功能的不可预测的结果。

[0798] 数据分块

[0799] I/O 设备记录的数据被划分为块。块的长度取决于设备;例如,块可以是卡、打印行或在磁带的两个相邻空隙之间记录的信息。

[0800] 在一个 I/O 操作中能被传送的最大数量的信息是一个块。当相关的主存储区域被耗尽或者达到块的结束时,不管哪个先出现,I/O 操作被终止。对于某些操作,例如在磁带单元或在询问站上写入,块未被定义,并且传送的信息量仅受程序控制。

[0801] 操作请求块

[0802] 操作 - 请求块(ORB)是 START SUBCHANNEL 的操作数。ORB 指定了在控制该特定开始功能时使用的参数。这些参数包括中断参数、子通道键、第一个 CCW 或 TCW 的地址、操作控制位、优先级控制号以及到指定设备的通道路径的逻辑可用性的指定。

[0803] 在执行 START SUBCHANNEL 时,在设置条件代码 0 之前,ORB 的内容被放在指定的子通道中。如果执行将带来非零的条件代码,ORB 的内容不会被放在指定的子通道中。

[0804] ORB 地址的两个最右侧的位必须为零,从而将 ORB 放在字边界上;否则,规格异常被识别。当光纤通道扩展(FCX)设施被安装,ORB 的通道程序类型控制(B)(字 1,位 13)指定了 ORB 所指定的通道程序的类型。当 B 是零,ORB 指定了 CCW 通道程序。当 B 是 1,ORB 指定了 TCW 通道程序。只有支持 FCX 的 I/O 设备才能识别 TCW 通道程序。

[0805] 在执行 START SUBCHANNEL 时,如果指定 CCW 通道程序的 ORB 的内容被放在指定的子通道中,子通道保持在命令模式下。于是,这样的 ORB 也被称为命令模式 ORB。在执行

START CHANNEL 时,如果指定 TCW 通道程序的 ORB 的内容被放在子通道中,则子通道进入传输模式。于是,这样的 ORB 也被称为传输模式 ORB。

[0806] TEST PENDING INTERRUPTION

[0807] 用于子通道上的待决 I/O 中断的 I/O 中断代码被存储在第二操作数地址指定的位置,且待决的 I/O 中断请求被清除。

[0808] 当第二操作数地址不是零时,它是存储由字 0 和 1 构成的双字 I/O 中断代码的位置的逻辑地址。第二操作数地址必须在字边界上指定;否则,规格异常被识别。

[0809] 如果第二操作数地址是零,由字 0-2 构成的三字 I/O 中断代码被存储在真实位置 184-195。在该情形下,不会应用低地址保护和键控制保护。

[0810] 在访问寄存器模式下当第二操作数地址是零时,无法预测访问寄存器 B2 是否发生访问寄存器转换。如果发生转换,产生的地址空间控制元素未被使用;即,中断代码仍被存储在真实位置 184-195。

[0811] 仅针对执行指令的 CPU 的控制寄存器 6 中的 I/O 中断子类掩码所允许的那些 I/O 中断子类来接受待决的 I/O 中断请求。如果不存在控制寄存器 6 所允许的 I/O 中断请求,I/O 中断代码不会被存储,第二操作数位置不会被修改,且条件代码 0 被设置。

[0812] 如果待决的 I/O 中断请求被接受,I/O 中断代码被存储,待决的 I/O 中断请求被清除,且条件代码 1 被设置。存储的 I/O 中断代码与出现 I/O 中断时将会存储的相同。但是,PSW 不像在出现 I/O 中断时那样被交换。指令的执行定义如下:

[0813] 子系统识别字(SID)

[0814] SID 的位 32-63 被放在字 0 中。

[0815] 中断参数:字 1 包括程序指定的四字参数,且被传递到 ORB 或 PMCW 的字 0 中。当设备呈现告警状态,且中断参数之前没有通过执行 START SUBCHANNEL 或 MODIFY SUBCHANNEL 而被传递到子通道,该字段包含零。

[0816] 中断识别字:当字 2 被存储时,它包含中断识别字,其还识别 I/O 中断的源。仅在第二操作数地址为零时才存储字 2。

[0817] 中断识别字被定义如下:

[0818] A 位(A):中断识别字的位 0 指定了被清除的待决 I/O 中断请求的类型。当位 0 是零,I/O 中断请求与子通道关联。

[0819] I/O 中断子类(ISC):中断识别字的位位置 2-4 包含无符号二进制整数,范围为 0-7,其指定了与子通道关联的 I/O 中断子类,针对该子通道待决中断请求被清除。剩余的位位置被保留并存储为零。

[0820] 特殊条件

[0821] TEST PENDING INTERRUPTION 会遇到如下描述或列出的程序异常。

[0822] 第二操作数必须在字边界上指定;否则,规格异常被识别。

[0823] TEST PENDING INTERRUPTION 的执行在所有寻址和保护异常上被抑制。

[0824] 产生的条件代码:

[0825] 0 中断代码未被存储

[0826] 1 中断代码被存储

[0827] 2—

[0828] 3—

[0829] 程序异常：

[0830] • 访问(存储,操作数2,第二操作地址仅为非零)

[0831] • 特许操作

[0832] • 规格

[0833] 编程说明

[0834] 1. 当 I/O 中断被屏蔽, TEST PENDING INTERRUPTION 应该仅以为零的第二操作数地址来执行。否则,如果出现 I/O 中断,指令存储的 I/O 中断代码会丢失。识别 TEST PENDING INTTERUPTION 之后出现的 I/O 中断的源的 I/O 中断代码也被存储在真实位置 184-195,代替由指令存储的 I/O 中断代码。

[0835] 2. 在访问寄存器模式下当第二操作数地址为零时,如果出现访问寄存器转换并且访问寄存器有错误,则访问异常被识别。可以通过使 B2 字段变成零或在访问寄存器中放置 00000000hex、00000001hex 或任何其他有效的内容来避免该异常。

[0836] STORE SUBCHANNEL

[0837] 指定的子通道的控制和状态信息被存储在指定的 SCHIB 中。

[0838] 通用寄存器 1 包含子系统识别字,其指定了要为其存储信息的子通道。第二操作数地址是 SCHIB 的逻辑地址且必须在字边界上指定;否则,规格异常被识别。

[0839] 当扩展 I/O 测量块设施未被安装时,在 SCHIB 中存储的信息由路径管理控制字、SCSC 以及依赖于模型的信息的三个字构成。当扩展 I/O 测量块设施已被安装,在 SCHIB 中存储的信息由路径管理控制字、SCSW、测量块地址字段以及依赖于模型的信息的一个字构成。

[0840] STORE SUBCHANNEL 的执行不会改变子通道处的任何信息。

[0841] 设置条件代码 0 以指示指定的子通道的控制和状态信息已被存储在 SCHIB 中。当 STORE SUBCHANNEL 的执行导致条件代码 0 的设置,SCHIB 中的信息指示子通道的一致状态。

[0842] 特殊条件

[0843] 当指定的子通道对于 STORE SUBCHANNEL 不可操作时,设置条件代码 3,且不采取其他动作。如果在通道子系统中未提供子通道,则子通道对于 STORE SUBCHANNEL 不可操作。

[0844] STORE SUBCHANNEL 会遇到以下描述或列出的程序异常。

[0845] 当多子通道组设施未被安装,通用寄存器 1 的位 32-47 必须包含 0001hex;否则,操作数异常被识别。

[0846] 当多子通道组设施已被安装,通用寄存器 1 的位 32-44 必须包含零,位 45-46 必须包含有效值,且位 47 必须包含值 1;否则,操作数异常被识别。

[0847] 第二操作数必须在字边界上指定;否则,规格异常被识别。

[0848] 产生的条件代码:

[0849] 0SCHIB 被存储

[0850] 1—

[0851] 2—

[0852] 3 不可操作

[0853] 程序异常

- [0854] • 访问(存储,操作数 2)
 - [0855] • 操作数
 - [0856] • 特许操作
 - [0857] • 规格
 - [0858] 编程说明
- [0859] 1. 在 SCSW 中存储的设备状态可以包括设备忙、控制单元忙或控制单元结束指示。
- [0860] 2. 从子通道获取在 SCHIB 中存储的信息。STORE SUBCHANNEL 指令不会使通道子系统询问被寻址的设备。
- [0861] 3. STORE SUBCHANNEL 可以任何时间执行,以对子通道中存在的条件进行抽样,而不会使任何待决状态条件被清除。
- [0862] 4. 应避免重复执行 STORE SUBCHANNEL 而没有中间的延迟(例如,以确定子通道何时改变状态),因为 CPU 对子通道的重复访问会延迟或阻止通道子系统访问子通道来更新该子通道。
- [0863] TEST SUBCHANNEL
- [0864] 子通道的控制和状态信息被存储在指定的 IRB 中。通用寄存器 1 包含子系统识别字,其指定了要为其存储信息的子通道。第二操作数地址是 IRB 的逻辑地址,且必须在字边界上指定;否则,规格异常被识别。
- [0865] 在 IRB 中存储的信息由 SCSW 字、扩展状态字和扩展控制字构成。
- [0866] 如果子通道是状态待决的,状态控制字段的状态待决位被存储为 1。子通道是否是状态待决的对执行 TEST SUBCHANNEL 时执行的功能具有影响。
- [0867] 当子通道是状态待决的并且 TEST SUBCHANNEL 被执行,如上所述,信息被存储在 IRB 中,接着清除在子通道中存在的特定条件和指示。如果子通道是在传输模式中,这些条件的清除特别是开始功能将子通道置于命令模式。如果 I/O 中断请求对于子通道是待决的,该请求被清除。设置条件代码 0 以指示已采取这些动作。
- [0868] 当子通道不是状态待决的并且 TEST SUBCHANNEL 被执行,信息被存储在 IRB 中,并且没有条件或指示被清除。设置条件代码 1 以指示已采用这些动作。
- [0869] 特殊条件
- [0870] 当子通道对于 TEST SUBCHANNEL 不可操作时,设置条件代码 3,且不采取其他动作。如果子通道未被提供、不具有与之相关的有效设备编号或未被启用,则子通道对于 TEST SUBCHANNEL 不可操作。
- [0871] TEST SUBCHANNEL 可以遇到如下描述或列出的程序异常。
- [0872] 当多子通道组设施未被安装,通用寄存器 1 的位 32-47 必须包含 0001hex;否则,操作数异常被识别。
- [0873] 当多子通道组设施已被安装,通用寄存器 1 的位 32-44 必须包含零,位 45-46 必须包含有效值,并且位 47 必须包含值 1;否则,操作数异常被识别。
- [0874] 第二操作数必须在字边界上指定;否则,规格异常被识别。
- [0875] 在在寻址和保护异常上结束 TEST SUBCHANNEL 的执行时,子通道的状态未被改变。
产生条件代码:
- [0876] 0IRB 被存储;子通道状态待决

- [0877] 1 IRB 被存储 ; 子通道不是状态待决
- [0878] 2—
- [0879] 3 不可操作
- [0880] 程序异常 :
- [0881] • 访问(存储、操作数 2)
 - [0882] • 操作数
 - [0883] • 特许操作
 - [0884] • 规格
- [0885] 编程说明
- [0886] 1. SCSW 中存储的设备状态可以包括设备忙、控制单元忙或控制单元结束的指示。
- [0887] 2. 从子通道获取 IRB 中存储的信息。TEST SUBCHANNEL 指令不会使通道子系统询问被寻址的设备。
- [0888] 3. 当出现 I/O 中断, 这是在子通道处的状态待决条件的结果, 且典型地执行 TEST SUBCHANNEL 来清除该状态。TEST SUBCHANNEL 也可以在任何其他时间执行以对子通道中存在的条件进行抽样。
- [0889] 4. 应避免重复执行 TEST SUBCHANNEL 来确定何时已完成开始功能, 因为存在某些条件, 在这些条件下可以指示或不指示开始功能的完成。例如, 如果由于另一子通道已经是状态待决的从而通道子系统保持接口控制检查(IFCC) 条件无效(abeyance) (针对任何子通道), 并且如果由 TEST SUBCHANNEL 测试的开始功能仅具有带有 IFCC 条件的通道路径作为仅有的可供选择的路径, 则可以不启动开始功能, 直到另一子通道中的状态待决条件被清除, 从而允许 IFCC 条件在其应用的子通道处被指示。
- [0890] 5. 应避免重复执行 TEST SUBCHANNEL 而没有中间的延迟例如来确定子通道何时改变状态, 因为 CPU 对子通道的重复访问会延迟或阻止通道子系统对子通道的访问。在几乎同一时间由多个 CPU 对同一子通道执行 TEST SUBCHANNEL 可以具有相同的效果, 并且也应被避免。
- [0891] 6. 可以通过执行 TEST SUBCHANNEL 来修改 CPU 进行的 I/O 中断处理的优先级。当 TEST SUBCHANNEL 被执行, 且指定的子通道具有 I/O 中断请求待决, 该 I/O 中断请求被清除, 且 SCSW 被存储, 而与之前建立的任何优先级无关。剩余的 I/O 中断请求的相对优先级不改变。
- [0892] 所属技术领域的技术人员知道, 本发明可以实现为系统、方法或计算机程序产品。因此, 本公开可以具体实现为以下形式, 即 : 可以是完全的硬件、也可以是完全的软件(包括固件、驻留软件、微代码等), 还可以是硬件和软件结合的形式, 本文一般称为“电路”、“模块”或“系统”。此外, 在一些实施例中, 本发明还可以实现为在一个或多个计算机可读介质中的计算机程序产品的形式, 该计算机可读介质中包含计算机可读的程序代码。
- [0893] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件, 或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括 : 具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储

器 (EPROM 或闪存)、光纤、便携式紧凑磁盘只读存储器 (CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0894] 现在参考图 10,在一个例子中,计算机程序产品 1000 包括,例如,一个或多个非易失性计算机可读存储介质 1002,在其上存储有计算机可读的程序代码装置或逻辑 1004,以提供并方便本发明的一个或多个方面。

[0895] 体现在计算机可读介质上的程序代码可以用任何适当的介质传输,所述介质包括但不限于:无线、有线、光缆、RF 等,或上述的任意合适的组合。

[0896] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言—诸如 Java、Smalltalk、C++,还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网 (LAN) 或广域网 (WAN) —连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0897] 本文中将参照本发明实施例的方法、装置(系统)和计算机程序产品的流程图和 / 或框图描述本发明。应当理解,流程图和 / 或框图的每个方框以及流程图和 / 或框图中各方框的组合,都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,这些计算机程序指令通过计算机或其它可编程数据处理装置执行,产生了实现流程图和 / 或框图中的方框中规定的功能 / 操作的装置。

[0898] 也可以把这些计算机程序指令存储在能使得计算机或其它可编程数据处理装置以特定方式工作的计算机可读介质中,这样,存储在计算机可读介质中的指令就产生出一个包括实现流程图和 / 或框图中的方框中规定的功能 / 操作的指令装置 (instruction means) 的制造品 (manufacture)。

[0899] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机或其它可编程装置上执行的指令能够提供实现流程图和 / 或框图中的方框中规定的功能 / 操作的过程。

[0900] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图和 / 或流程图中的每个方框、以及框图和 / 或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0901] 除了上述,本发明的一个或多个方面可由服务提供商提供、许诺(offer)、部署、管理、服务等,该服务提供商提供用户环境的管理。例如,服务提供商可创建、维持、支持等计算机代码和 / 或计算机基础设施,其为一个或多个用户执行本发明的一个或多个方面。反过来,服务提供商可例如根据预订和 / 或费用协议从用户接受付费。额外地或可替换地,服务提供商可从向一个或多个第三方销售广告内容接受付费。

[0902] 在本发明的一个方面,可部署用于执行本发明的一个或多个方面的应用。作为一个例子,部署应用包括提供计算机基础设施,其可操作以执行本发明的一个或多个方面。

[0903] 作为本发明的另一个方面,可部署计算基础设施,其包括将计算机可读代码集成到计算机系统,其中与计算系统结合的代码能够执行本发明的一个或多个方面。

[0904] 作为本发明的再一个方面,可提供用于集成计算基础设施包括将计算机可读码集成到计算机系统的过程。计算机系统包括计算机可读介质,其中计算机介质包括本发明的一个或多个方面。与计算机系统结合的代码能够执行本发明的一个或多个方面。

[0905] 尽管以上描述了各种实施例,这些仅是示例。例如,其他架构的计算环境可包含和使用本发明的一个或多个方面。作为示例,z196 服务器之外的服务器可包括、使用和 / 或受益于本发明的一个或多个方面。此外,其他指令和 / 或命令可被使用;且指令和 / 或命令可包括与这里描述的相比附加的、更少的和 / 或不同的信息。许多变化是可能的。

[0906] 而且,其他类型的计算环境可受益于本发明的一个或多个方面。作为例子,可使用适于存储和 / 或执行程序代码的数据处理系统,其包括至少两个通过系统总线直接或间接耦合到存储元件的处理器。存储器元件包括,例如,在程序代码的实际执行期间使用的本地存储器、大容量存储器以及高速缓冲存储器,其提供至少一些程序代码的临时存储,以便减少在执行期间必须从大容量存储器取回代码的次数。

[0907] 输入 / 输出或 I/O 设备(包括但不限于键盘、显示器、指点设备、DASD、磁带、CD、DVD、拇指驱动器(thumb drive)以及其他存储介质等)可直接或通过介于其间的 I/O 控制器被耦合到系统。网络适配器也可被耦合到系统以使得数据处理系统能够通过介于其间的私有或公共网络而耦合到其他的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仅是一些可获得的网络适配器类型。

[0908] 参考图 11,其描述了实施本发明的一个或多个方面的主机计算机系统 5000 的代表性组件。代表性主机计算机 5000 包括与计算机存储器(即,中央存储器)5002 通信的一个或多个 CPU5001,以及到存储介质设备 5011 和网络 5010 的 I/O 接口,以用于与其他计算机或 SAN 等通信。CPU5001 符合具有架构指令集和架构功能的架构。CPU5001 可具有动态地址转换(DAT)5003,其用于将程序地址(虚拟地址)转变为存储器的真实地址。DAT 典型地包括用于高速缓存转换的转换后备缓冲器(TLB)5007,这样稍后对计算机存储器 5002 块的访问不需要地址转换的延迟。典型地,高速缓存 5009 被使用在计算机存储器 5002 和处理器 5001 之间。高速缓存 5009 可以是分层的,具有可被多于一个的 CPU 获得的大高速缓存,以及大高速缓存和每个 CPU 之间的较小、较快(较低级)的高速缓存。在一些实施方式中,较低级的高速缓存被拆分以为指令获取和数据访问提供单独的低级高速缓存。在一个实施例中,由指令获取单元 5004 经由高速缓存 5009 从存储器 5002 获取指令。指令在指令解码单元 5006 中被解码,且(在一些实施例中与其他指令一起)被发送到一个或多个指令执行单元 5008。典型地,使用若干执行单元 5008,例如算术执行单元、浮点执行单元和分支指令执行

单元。指令被执行单元执行,如需要,从指令指定的寄存器或存储器访问操作数。如果将从存储器 5002 访问(加载或存储)操作数,加载 / 存储单元 5005 典型地在被执行的指令的控制下处理该访问。指令可在硬件电路或内部微代码(固件)中或其组合中被执行。

[0909] 需注意的是,计算机系统包括本地(或主)存储器中的信息,以及寻址、保护以及引用和改变记录。寻址的一些方面包括地址格式、地址空间的概念、地址的各种类型和其中一种类型的地址被转换为另一种类型的地址的方式。一些主存储器包括永久分配的存储位置。主存储器向系统提供直接可被寻址的快速访问的数据存储。数据和程序在可被处理之前都将(从输入设备)被加载到主存储器。

[0910] 主存储器可包括一个或多个更小、更快速访问的缓冲存储器,有时候被称为高速缓存。高速缓存典型地与 CPU 或 I/O 处理器物理地关联。物理结构以及不同存储介质的使用的效果,除了在性能上,通常不会被程序观察到。

[0911] 可维护用于指令和数据操作数的单独的高速缓存。高速缓存中的信息可被维护为相邻的字节,所述字节位于被称为高速缓存块或高速缓存线(或简称为线)的整数界限上。模型可提供 EXTRACT CACHE ATTRIBUTE(提取高速缓存属性)指令,其返回高速缓存线的字节大小。模型也可提供 PREFETCH DATA (预取数据)和 PREFETCH DATA RELATIVE LONG (预取较长数据) 指令,其实现存储到数据或指令高速缓存中的预取,或数据从高速缓存的释放。

[0912] 存储器被视为位的长水平串。对于大部分操作来说,以从左到右的顺序进行对存储器的访问。位串被细分为八个位的单位。八位单位被称为字节,其是所有信息格式的基本构件。存储器中的每个字节位置由唯一的非负整数标识,该非负整数是该字节位置的地址,或简称为字节地址。相邻的字节位置具有连续的地址,在左边从 0 开始且以从左到右的顺序进行。地址是无符号二进制整数,且是 24、31 或 64 位。

[0913] 信息一次一个字节或一组字节地在存储器和 CPU 或通道子系统之间传递。除非另有指定,例如在 z/Architecture O R 中,存储器中的一组字节由该组的最左边的字节寻址。组中的字节的数量可由将被执行的操作暗示或显式地指定。当在 CPU 操作中使用时,一组字节被称为字段。在字节的每个组内,例如在 z/Architecture O R 中,位以从左到右的顺序被编号。在 z/Architecture O R 中,最左边的位有时候被称为“高阶”位且最右边的位被称为“低阶”位。但是,位数不是存储器地址。仅字节可被寻址。为了操作存储器中的字节的单个位,访问整个字节。字节上的位从左到右被编号为 0 到 7(例如在 z/Architecture O R 中)。对于 24 位地址,地址中的位被编号为 8-31 或 40-63,对于 31 位地址,编号为 1-31 或 33-63,对于 64 位地址,编号为 0-63。在多个字节的任何其他的固定长度的格式中,构成格式的位从 0 开始被连续编号。为了错误检测,且优选地为了校正,一个或多个校验位可与每一个字节或一组字节一起被传递。这样的校验位由机器自动生成且不能被程序直接控制。存储容量以字节的数量来表示。当存储器操作数字段的长度由指令的操作码暗示时,字段被称为具有固定长度,其可以是一个、两个、四个、八个或十六个字节。可为某些指令暗示更大的字段。当存储器操作数字段的长度没有被暗示,而是被显式地表示时,该字段被称为具有可变长度。可变长度的操作数可以一个字节的增量(或者对于一些指令,以两个字节倍数或其他倍数)在长度上可变。当信息被放在存储器中时,仅替换被包括在指定的字段中的那些字节位置的内容,即使到存储器的物理路径的宽度可能大于正被存储的字段的长度。

[0914] 某些信息单元位于存储器中的整数界限上。对于信息单元,当其存储器地址是以

字节表示的单元长度的倍数时,界限被称为是整数的。特殊的名称被给予整数界限上的 2、4、8 和 16 字节的字段。半字是两字节边界上的一组两个连续的字节,且是指令的基本构件。字是四字节边界上的一组四个连续的字节。双字是八字节边界上一组八个连续的字节。四倍长字(quadword)是 16 字节边界上的一组 16 个连续的字节。当存储器地址指定半字、字、双字和四倍长字时,地址的二进制表示分别包括一个、两个、三个或四个最右边的零位。指令将位于二字节整数边界上。大多数指令的存储器操作数不具有界限对准要求。

[0915] 在为指令和数据操作数实现单独的高速缓存的设备上,如果程序在高速缓存线中存储且指令被随后从该高速缓存线获取,可经历显著的延迟,不管该存储是否改变随后被获取的指令。

[0916] 在一个实施例中,本发明可被软件(有时候被称为许可的内部代码、固件、微代码、毫代码、微微代码(pico-code)等,其任何一个都将符合本发明)实施。参考图 11,体现本发明的软件程序代码可典型地由主系统 5000 的处理器 5001 从长期存储介质设备 5011(诸如 CD-ROM 驱动器、磁带驱动或硬盘驱动器)访问。软件程序代码可体现在与数据处理系统一起使用的各种已知介质(诸如软磁盘、硬盘驱动或 CD-ROM)中的任何一个上。代码可在这样的介质上被分发,或可从一个计算机系统的计算机存储器 5002 或存储设备通过网络 5010 被分发给其他计算机系统的用户,以由这样的其他系统的用户使用。

[0917] 软件程序代码包括操作系统,其控制各种计算机组件和一个或多个应用程序的功能和交互。程序代码通常可从存储介质设备 5011 调页到相对更高速的计算机存储器 5002,在此它对于处理器 5001 是可用的。用于在存储器中、物理介质上体现软件程序代码和 / 或经由网络分发软件代码的技术和方法是熟知的,且不会在此被进一步讨论。当程序代码被创建并存储在有形介质(包括但不限于电子存储模块(RAM)、闪存、光盘(CD)、DVD、磁带等)上时,其经常被称为“计算机程序产品”。计算机程序产品介质典型地可由优选地位于计算机系统中的处理电路读取以由处理电路执行。

[0918] 图 12 示出了可在其中实施本发明的代表性工作站或服务器硬件系统。图 12 的系统 5020 包括代表性基本计算机系统(base computer system)5021,诸如个人计算机、工作站或服务器,包括可选的外围设备。根据已知技术,基本计算机系统 5021 包括一个或多个处理器 5026 以及被用于连接并使能处理器 5026 和系统 5021 的其他组件之间的通信的总线。总线将处理器 5026 连接到存储器 5025 以及可包括例如硬盘驱动器(例如,包括磁介质、CD、DVD 和闪存中的任何一个)或磁带驱动器的长期存储器 5027。系统 5021 也可包括用户接口适配器,其经由总线将微处理器 5026 连接到一个或多个接口设备,诸如键盘 5024、鼠标 5023、打印机 / 扫描仪 5030 和 / 或其他接口设备,其可以是任何用户接口设备,诸如触摸敏感屏、数字化输入垫(digitized entry pad)等。总线也可经由显示适配器将诸如 LCD 屏幕或监视器的显示设备 5022 连接到微处理器 5026。

[0919] 系统 5021 可通过能与网络 5029 通信 5028 的网络适配器与其他计算机或计算机网络通信。示例性网络适配器是通信通道、令牌环网、以太网或调制解调器。或者,系统 5021 可使用诸如 CDPD(蜂窝数字分组数据)卡的无线接口来通信。系统 5021 可与局域网(LAN)或广域网(WAN)中的这样的其他计算机关联,或系统 5021 可以是与另一个计算机的客户机 / 服务器安排中的客户机等。所有这些配置以及合适的通信硬件和软件在本领域中是已知的。

[0920] 图 13 示出了其中可实施本发明的数据处理网络 5040。数据处理网络 5040 可包括多个单独的网络,诸如无线网和有线网,其每个可包括多个单独的工作站 5041、5042、5043、5044。此外,本领域技术人员将理解,可包括一个或多个 LAN,其中 LAN 可包括多个耦合到主处理机的智能工作站。

[0921] 仍然参考图 13,网络也可包括大型计算机或服务器,诸如网关计算机(客户机服务器 5046)或应用服务器(远程服务器 5048,其可访问数据储存库,且也可直接从工作站 5045 被访问)。网关计算机 5046 用作到每个单独网络的进入点。当将一个联网协议连接到另一个时,需要网关。网关 5046 可通过通信链路优选地耦合到另一个网络(例如因特网 5047)。也可使用通信链路将网关 5046 直接耦合到一个或多个工作站 5041、5042、5043、5044。可以利用可从国际商业机器公司获得的 IBM eServerTMSystem z O R 服务器来实现网关计算机。

[0922] 同时参考图 12 和 13,可体现本发明的软件编程代码可被系统 5020 的处理器 5026 从诸如 CD-ROM 驱动器或硬盘驱动器的长期存储介质 5027 访问。软件编程代码可被体现在与数据处理系统一起使用的各种已知介质(诸如软盘、硬盘驱动器或 CD-ROM)中的任一个上。代码可在这样的介质上被分发,或从一个计算机系统的存储器或存储设备通过网络被分发到其他计算机系统的用户 5050、5051,以供这样的其他系统的用户使用。

[0923] 或者,编程代码可体现在存储器 5025 中,且由处理器 5026 使用处理器总线访问。这样的编程代码包括操作系统,其控制各种计算机组件和一个或多个应用程序 5032 的功能和交互。程序代码通常从存储介质 5027 调页到高速存储器 5025,在此它可用于由处理器 5026 进行处理。用于在存储器中、在物理介质上体现软件编程代码和 / 或经由网络分发软件代码的技术和方法是公知的,不会在此进一步讨论。程序代码,当其被创建且在有形介质(包括但不限于电子存储模块(RAM)、闪存、光盘(CD)、DVD、磁带等)上存储时,通常被称为“计算机程序产品”。计算机程序产品介质典型地可以被优选地位于计算机系统中的处理电路读取以由处理电路执行。

[0924] 最容易被处理器使用的高速缓存(通常比处理器的其他高速缓存更快更小)是最顶级(L1 或级别 1)高速缓存,且主存储(主存储器)是最高级高速缓存(如果有三个级别的話是 L3)。最低级高速缓存经常被分为保持将被执行的机器指令的指令缓存(I- 高速缓存),和保持数据操作数的数据高速缓存(D- 高速缓存)。

[0925] 参考图 14,为处理器 5026 示出了示例性处理器实施例。典型地,使用一个或多个级别的高速缓存 5053 来缓冲存储器块,以便改善处理器性能。高速缓存 5053 是高速缓冲器,其保持很可能被使用的存储器数据的高速缓存线。典型的高速缓存线是 64、128 或 256 字节的存储器数据。通常使用单独的高速缓存以用于缓存指令而不是缓存数据。高速缓存一致性(存储器和高速缓存中的线的副本的同步)通常由本领域中熟知的各种“窥探”算法提供。处理器系统的主存储器 5025 通常被称为高速缓存。在具有 4 个级别的高速缓存 5053 的处理器系统中,主存储器 5025 有时候被称为级别 5 (L5) 高速缓存,因为它典型地更快,且仅保持可被计算机系统使用的非易失性存储器(DASD、磁带等)的一部分。主存储器 5025 可“高速缓存”由操作系统向主存储器 5025 调页入或从其调页出的数据页。

[0926] 程序计数器(指令计数器)5061 保持跟踪将被执行的当前指令的地址。z/Architecture O R 处理器中的程序计数器是 64 位的,且可被截短为 31 或 24 位以支持先前

的寻址界限。程序计数器典型地体现在计算机的 PSW (程序状态字) 中,这样它可在上下文转换中持续。因此,具有程序计数值的进行中的程序可被例如操作系统中断(从程序环境到操作系统环境的上下文转换)。当程序不活动时,程序的 PSW 维持程序计数值,且在操作系统执行时,操作系统的(PSW 中的)程序计数器被使用。典型地,程序计数器以等于当前指令的字节数的量增量。RISC (精简指令集计算) 指令典型地是固定长度,而 CISC (复杂指令集计算) 指令典型地是可变长度。IBMz/Architecture O R 的指令是具有长度为 2、4 或 6 字节的 CISC 指令。程序计数器 5061 被例如上下文转换操作或分支指令的分支采取操作修改。在上下文转换操作中,当前的程序计数值与关于正被执行的程序的其他状态信息(诸如条件码)一起被保存在程序状态字中,且新程序计数值被载入并指向将被执行的新程序模块的指令。执行分支采取操作,以通过将分支指令的结果加载到程序计数器 5061 中而允许程序进行决定或在程序内循环。

[0927] 典型地,使用指令获取单元 5055 代表处理器 5026 获取指令。获取单元可获取“下一序列指令”、分支采取指令的目标指令或上下文转换后的程序的第一指令。现在的指令获取单元通常使用预取技术基于被预取的指令将被使用的可能性来推测性地预取指令。例如,获取单元可获取 16 字节的指令,其包括下一顺序指令以及进一步的顺序指令的额外字节。

[0928] 获取的指令随后被处理器 5026 执行。在一实施例中,获取的指令被传递给获取单元的分派单元 5056。分派单元解码指令并将关于解码的指令的信息转送给合适的单元 5057、5058、5060。执行单元 5057 将典型地从指令获取单元 5055 接收关于解码的算术指令的信息,并将根据指令的操作码对操作数执行算术操作。优选地从存储器 5025、架构寄存器 5059 或从正被执行的指令的立即字段(immediate field)向执行单元 5057 提供操作数。执行的结果,当被存储时,被存储在存储器 5025、寄存器 5059 或其他机器硬件(诸如控制寄存器、PSW 寄存器等)中。

[0929] 处理器 5026 典型地具有一个或多个用于执行指令的功能的单元 5057、5058、5060。参考图 15A,执行单元 5057 可通过接口逻辑 5071 与架构通用寄存器 5059、解码 / 分派单元 5056、加载存储单元 5060 和其他 5065 处理器单元通信。执行单元 5057 可使用几个寄存器电路 5067、5068、5069 来保持算术逻辑单元(ALU) 5066 将操作的信息。ALU 执行诸如加减乘除的算术操作,以及诸如和、或以及异或(XOR)、旋转和移位的逻辑运算。优选地,ALU 支持依赖于设计的专门操作。其他电路可提供其他架构工具 5072,例如包括条件码和恢复支持逻辑。典型地,ALU 操作的结果被保持在输出寄存电路 5070 中,该输出寄存器电路可将结果转送到多种其他处理功能。有许多处理器单元安排,本说明书仅旨在提供对一个实施例的代表性理解。

[0930] 例如,ADD 指令将在具有算术和逻辑功能的执行单元 5057 中被执行,而例如浮点指令将在具有专用浮点能力的浮点执行中被执行。优选地,执行单元通过在操作数上执行操作码定义的功能在由指令标识的操作数上操作。例如,ADD 指令可被执行单元 5057 在由指令的寄存器字段标识的两个寄存器 5059 中发现的操作数上执行。

[0931] 执行单元 5057 对两个操作数执行算术加法,并在第三操作数中存储结果,其中第三操作数可以是第三寄存器或两个源寄存器中的一个。执行单元优选地利用算术逻辑单元(ALU) 5066,其能执行多种逻辑功能,诸如移位、旋转、和、或、异或,以及多种代数函数,包

括加减乘除中的任何一个。有些 ALU5066 被设计为用于标量运算,有些用于浮点。根据架构,数据可以是大端(big endien) (其中最低有效字节位于最高字节地址)或小端(little endien) (其中最低有效字节位于最低字节地址)。IBMz/Architecture O R 是大端。根据架构,带符号字段可以是符号和幅度、1 的补码或 2 的补码。2 的补码数是有利的,其在于 ALU 不需要设计减法能力,因为不管是 2 的补码中的负值还是正值,都仅要求 ALU 中的加法。数字通常以速记描述,其中 12 位的字段定义了 4096 字节块的地址,且通常被描述为例如 4Kbyte (千字节) 块。

[0932] 参考图 15B,用于执行分支指令的分支指令信息典型地被发送到分支单元 5058,该分支单元经常使用诸如分支历史表 5082 的分支预测算法,在其他条件运算完成前预测分支结果。在条件运算完成前,当前分支指令的目标将被获取并推测性地执行。当条件运算完成时,基于条件运算的条件和推测的结果,推测性执行的分支指令或被完成或被丢弃。典型的分支指令可测试条件码,以及如果条件码满足分支指令的分支要求,分支到目标地址,分支地址可基于若干数被计算,所述数包括例如在寄存器字段或是指令的立即字段中找到的数。分支单元 5058 可利用具有多个输入寄存器电路 5075、5076、5077 和一个输出寄存器电路 5080 的 ALU5074。分支单元 5058 可与例如通用寄存器 5059、解码分派单元 5056 或其他电路 5073 通信。

[0933] 一组指令的执行可由于多个原因中断,所述原因包括例如由操作系统发起的上下文转换、引起上下文转换的程序异常或错误、引起上下文转换的 I/O 中断信号或多个程序(在多线程环境中)的多线程活动。优选地,上下文转换动作保存关于当前执行的程序的状态信息,且随后加载关于正被调用的另一个程序的状态信息。状态信息可被存储在例如硬件寄存器或存储器中。状态信息优选地包括指向将被执行的下一个指令的程序计数器值、条件码、存储器转换信息和架构寄存器内容。上下文转换活动可被硬件电路、应用程序、操作系统程序或固件代码(微代码、微微代码或许可内部码(LIC))单独地或其组合实现。

[0934] 处理器根据指令定义的方法而访问操作数。指令可使用指令的一部分的值提供立即操作数,可提供一个或多个寄存器字段,其显式地指向通用寄存器或专用寄存器(例如浮点寄存器)。指令可利用由操作码字段确定的暗示的寄存器作为操作数。指令可利用用于操作数的存储器位置。可由寄存器、立即字段或寄存器和立即字段的组合提供操作数的存储器位置,如由 z/Architecture O R 长位移工具(facility)所示例的,其中该指令定义了基寄存器、索引寄存器和立即字段(位移字段),它们加到一起,以提供例如存储器中的操作数的地址。除非另外指明,此处的位置典型地意味着主存储器(主存储设备)中的位置。

[0935] 参考图 16C,处理器使用加载 / 存储单元 5060 访问存储器。加载 / 存储单元 5060 可以通过获取存储器 5053 中的目标操作数的地址并将操作数加载到寄存器 5059 或其他存储器 5053 位置中,来执行加载操作,或可以通过获取存储器 5053 中的目标操作数的地址并将从寄存器 5059 或另一个存储器 5053 位置获得的数据存储在存储器 5053 中的目标操作数位置,来执行存储操作。加载 / 存储单元 5060 可以是推测性的,且可以相对于指令顺序来说无序的顺序访问存储器,但是加载 / 存储单元 5060 将向程序维持指令按顺序执行的外观。加载 / 存储单元 5060 可与通用寄存器 5059、解密 / 分派单元 5056、高速缓存 / 存储器接口 5053 或其他元件 5083 通信,且包括各种寄存器电路、ALU5085 和控制逻辑 5090 以计算存储器地址并提供流水线顺序以使操作保持次序。一些操作可不按顺序,但加载 / 存

储单元提供功能以使不按顺序执行的操作对程序看起来如已按顺序执行一样,如本领域所熟知的。

[0936] 优选地,应用程序“看到的”地址通常被称为虚拟地址。虚拟地址有时候被称为“逻辑地址”和“有效地址”。这些虚拟地址之所以虚拟,在于它们由多种动态地址转换(DAT)技术中的一种重定向到物理存储器位置,所述动态地址转换技术包括但不限于简单地给用偏移值给虚拟地址加前缀、经由一个或多个转换表转换虚拟地址,所述转换表优选地包括至少一个段表和一个页表(单独地或组合地),优选地,段表具有指向页表的项。在z/Architecture O R中,提供转换分级结构,包括区域第一表、区域第二表、区域第三表、段表和可选的页表。地址转换的性能通常通过利用转换后备缓冲器(TLB)被改善,该转换后备缓冲器包括将虚拟地址映射到相关的物理存储位置的项。当DAT使用转换表转换虚拟地址时,创建项。于是,虚拟地址的随后使用可利用快的TLB的项,而不是慢的顺序转换表访问。TLB内容可由包括LRU(最少最近使用)的多个替换算法来管理。

[0937] 在处理器是多处理器系统的处理器的情况下,每个处理器具有保持共享资源的责任,所述共享资源诸如I/O、高速缓存、TLB和存储器,它们互锁以实现一致性。典型地,“窥探”技术将被用于维持高速缓存一致性。在窥探环境中,每个高速缓存线可被标记为正处于共享状态、独占状态、改变状态、无效状态等中的一个,以便有助于共享。

[0938] I/O单元5054(图14)向处理器提供用于附加到例如包括磁带、盘、打印机、显示器和网络的外围设备的装置。I/O单元通常由软件驱动器向计算机程序呈现。在诸如来自IBM O R的System z O R的大型计算机中,通道适配器和开放系统适配器是提供操作系统和外围设备之间的通信的大型计算机的I/O单元。

[0939] 而且,其他类型的计算环境可受益于本发明的一个或多个方面。作为例子,环境可包括仿真器(例如,软件或其他仿真机制),其中特定架构(包括例如指令执行、诸如地址转换的架构功能、以及架构寄存器)或其子集被仿真(例如,在具有处理器和存储器的本机计算机系统中)。在这样的环境中,仿真器的一个或多个仿真功能可实施本发明的一个或多个方面,即使执行仿真器的计算机可具有与正被仿真的能力不同的架构。作为一个例子,在仿真模式中,解码正被仿真的特定指令或操作,且建立合适的仿真功能以实施单个指令或操作。

[0940] 在仿真环境中,主计算机包括例如存储器以存储指令和数据;指令获取单元以从存储器获取指令,且可选地,提供用于获取的指令的本地缓冲;指令解码单元以接收取获的指令并确定已被获取的指令的类型;以及指令执行单元以执行该指令。执行可包括将数据从存储器加载到寄存器;从寄存器将数据存储回存储器;或执行如由解码单元确定的某些类型的算术或逻辑运算。在一个例子中,每个单元在软件中实现。例如,被所述单元执行的操作被实现为仿真器软件中的一个或多个子例程。

[0941] 更具体地,在大型计算机中,程序员(通常是如今的“C”程序员)一般通过编译器应用使用架构机器指令。存储在存储介质中的这些指令可以在z/Architecture O RIBM O R服务器中本机地执行,或在执行其他架构的机器中执行。它们可在现有的和未来的IBM O R大型计算机服务器以及IBM O R的其他机器(例如,Power Systems服务器和System x O R服务器)中被仿真。它们可在使用由IBM O R、Intel O R、AMDTM等制造的硬件的各种机器上运行Linux的机器中被执行。除了在z/Architecture O R下的该硬件上执行,Linux

也可被用于这样的机器,其使用由 TurboHercules、Hercules、UMX 或 FSI (Fundamental Software, Inc) (其中一般地执行是处于仿真模式中) 提供的仿真。在仿真模式中,仿真软件由本机处理器执行以仿真被仿真处理器的架构。

[0942] 本机处理器典型地执行仿真软件,其包括固件或本机操作系统,以执行被仿真处理器的仿真程序。仿真软件负责获取并执行被仿真处理器架构的指令。仿真软件维护仿真的程序计数器以保持跟踪指令界限。仿真软件可一次获取一个或多个仿真的机器指令,并将所述一个或多个仿真的机器指令转换为对应的本机机器指令组,以由本机处理器执行。这些转换的指令可被高速缓存,这样可完成更快的转换。仿真软件将维持被仿真的处理器架构的架构规则以保证为被仿真处理器编写的操作系统和应用正确操作。而且,仿真软件将提供由被仿真的处理器架构确定的资源,包括但不限于控制寄存器、通用寄存器、浮点寄存器、例如包括段表和页表的动态地址转换功能、中断机制、上下文转换机制、日中时间(TOD)时钟和到 I/O 子系统的架构接口,这样被设计为在被仿真处理器上运行的操作系统或应用程序可在具有仿真软件的本机处理器上运行。

[0943] 解码正被仿真的特定指令,且调用子例程以执行该单个指令的功能。仿真被仿真处理器的功能的仿真软件功能例如在“C”子例程或驱动器中实现,或由提供用于特定硬件的驱动器的其他方法实现,如本领域技术人员在理解优选实施例的描述后将理解的。包括但不限于 Beausoleil 等人的标题为“Multiprocessor for Hardware Emulation”的美国专利证书号 5,551,013 ;以及 Scalzi 等人的标题为“Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor”的美国专利证书号 6,009,261 ;以及 Davidian 等人的标题为“Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions”的美国专利证书号 5,574,873 ;以及 Gorishek 等人的标题为“Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System”的美国专利证书号 6,308,255 ;以及 Lethin 等人的标题为“Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method”的美国专利证书号 6,463,582, ;以及 Eric Traut 的标题为“Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions”的美国专利证书号 5,790,825 ;以及许多其他专利的各种软件和硬件仿真专利示出各种已知的方式来实现针对可为本领域技术人员获得的目标机器对为不同机器进行架构设计的指令格式的仿真,上述专利在这里都被作为参考。

[0944] 在图 16 中,提供了仿真主计算机系统 5092 的例子,其仿真主架构的主计算机系统 5000'。在仿真主计算机系统 5092 中,主处理器(CPU) 5091 是仿真主处理器(或虚拟主处理器),并包括具有与主计算机 5000' 的处理器 5091 不同的本机指令集架构的仿真处理器 5093。仿真主计算机系统 5092 具有可被仿真处理器 5093 访问的存储器 5094。在示例性实施例中,存储器 5094 被分区为主计算机存储器 5096 部分和仿真例程 5097 部分。根据主计算机架构,主计算机存储器 5096 对于仿真主计算机 5092 的程序来说是可用的。仿真处理器 5093 执行与被仿真处理器 5091 不同架构的架构指令集的本机指令(即来自仿真程序处理器 5097 的本机指令),且可通过使用从顺序和访问 / 解码例程获得的一个或多个指令从主计算机存储器 5096 中的程序访问用于执行的主机指令,所述顺序和访问 / 解码例程可解

码访问的主机指令,以确定用于仿真被访问的主机指令的功能的本机指令执行例程。被定义用于主计算机系统 5000' 架构的其他工具可被架构工具例程仿真,所述架构工具例程包括诸如通用寄存器、控制寄存器、动态地址转换和 I/O 子系统支持和处理器高速缓存等工具。仿真例程也可利用在仿真处理器 5093 中可获得的功能(诸如通用寄存器和虚拟地址的动态转换)以改善仿真例程的性能。也可提供专用硬件和卸载引擎以辅助处理器 5093 来仿真主计算机 5000' 的功能。

[0945] 在此使用的术语仅是为了描述特定实施例,且不旨在限制本发明。如在此使用的,单数形式“一”、“一个”和“该”也旨在包括复数形式,除非上下文另外清楚地指明。还将理解,当在说明书中使用时,术语“包括”和 / 或“包含”指明存在所述的特征、整体、步骤、操作、元件和 / 或组件,但不排除存在或附加一个或多个其他特征、整体、步骤、操作、元件和 / 或组件。

[0946] 所附权利要求书中的所有设备或步骤加功能元件的相应结构、材料、操作以及等价物,如有的话,旨在包括用于结合如特别要求保护的其他所要求保护的元件来执行所述功能的任何结构、材料或操作。呈现本发明的说明是为了示出和描述的作用,但不是穷尽性的或将本发明限制于所公开的形式。许多修改和变化对本领域普通技术人员来说是明显的,且不脱离本发明的范围和精神。选择和描述实施例是为了最佳地解释本发明的原理和实际应用,并使得本领域普通技术人员能针对适于考虑的特定用途的具有各种修改的各种实施例理解本发明。

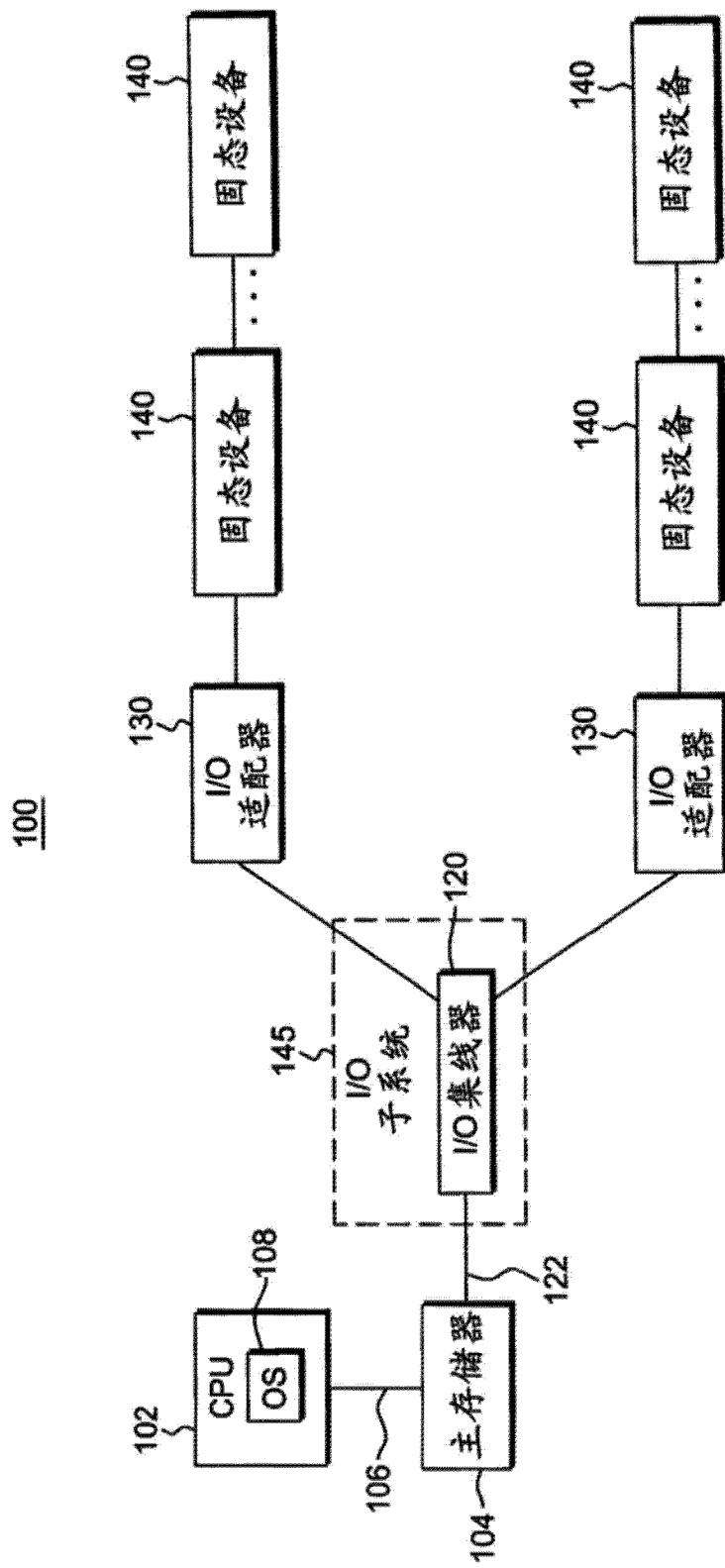


图 1A

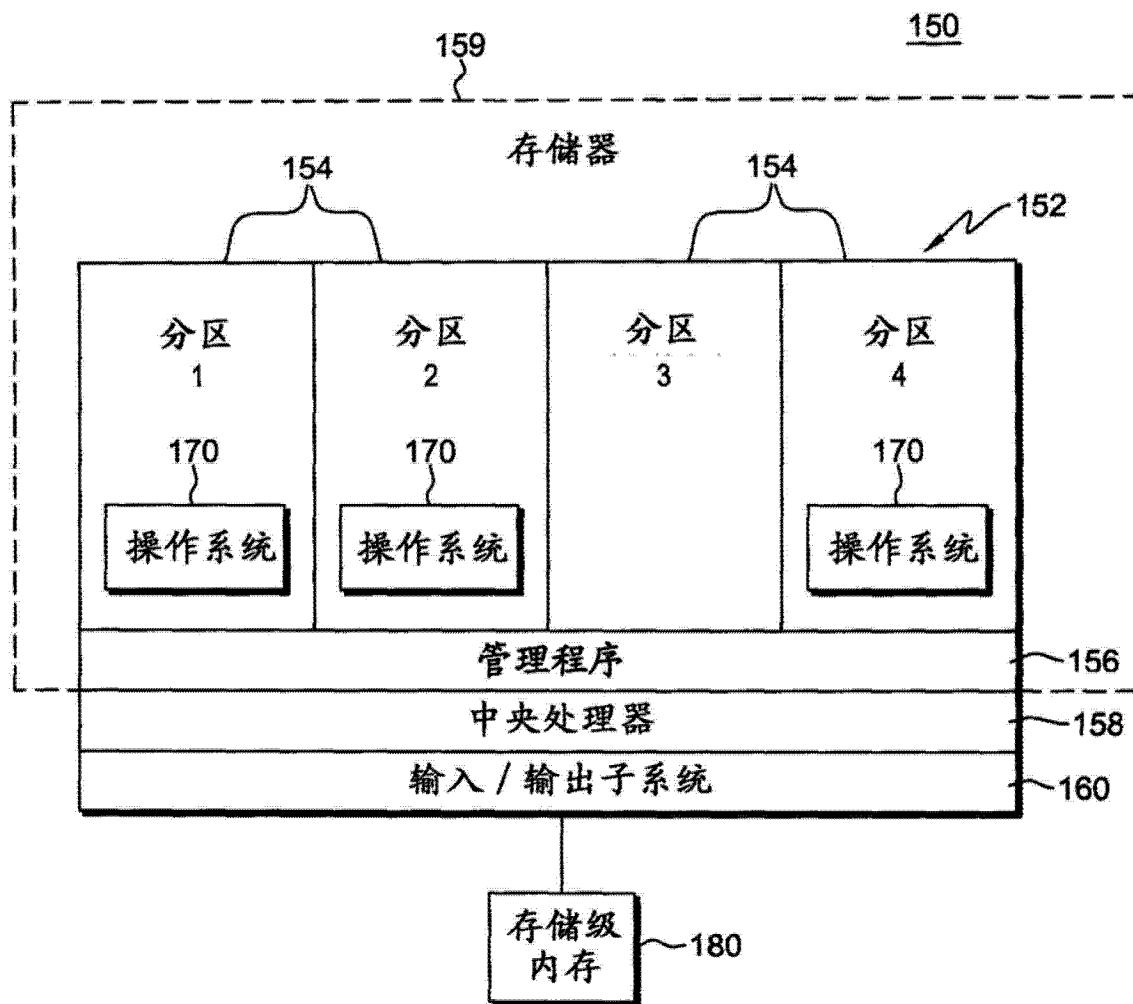


图 1B

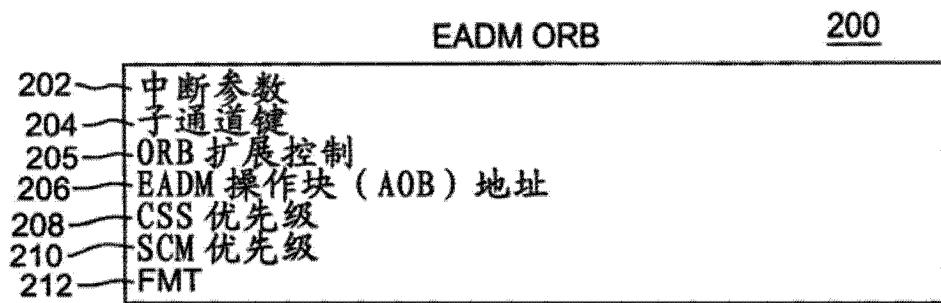


图 2A

EADM 操作块 (AOB)

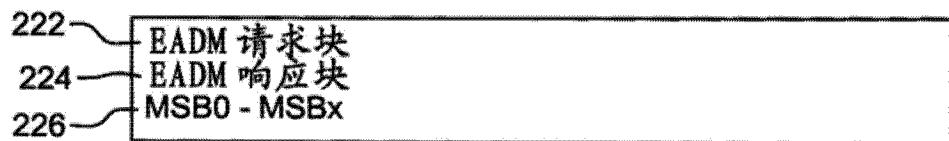
220

图 2B

EADM 请求块

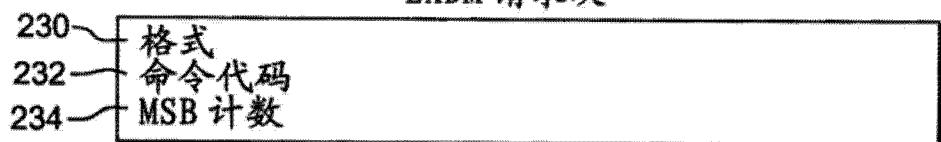
222

图 2C

EADM 响应块

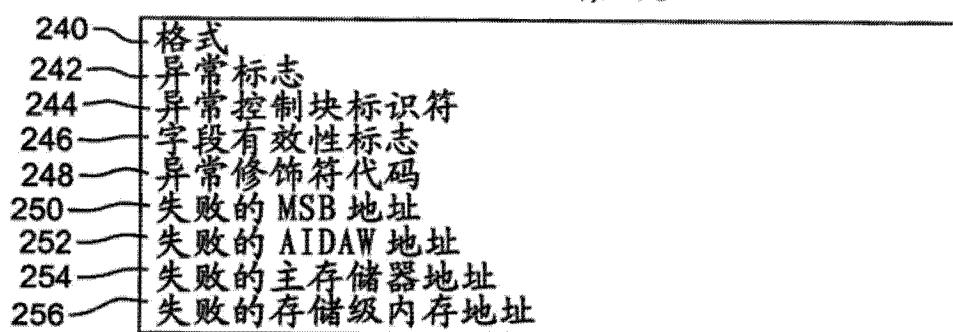
224

图 2D

EADM 移动规格块

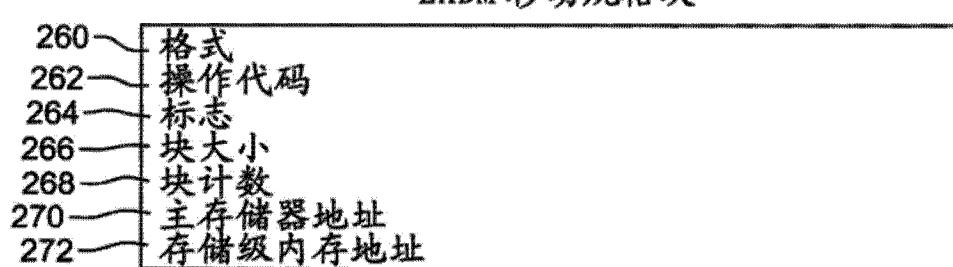
226

图 2E

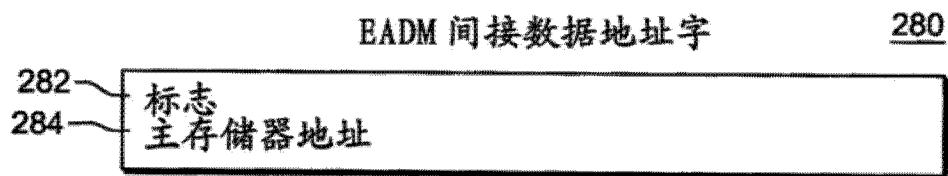


图 2F

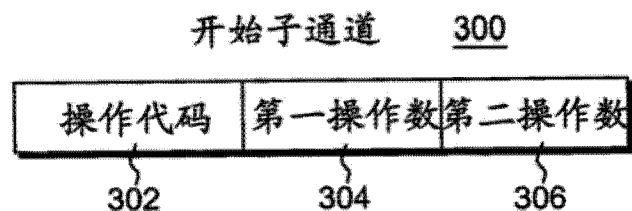


图 3A

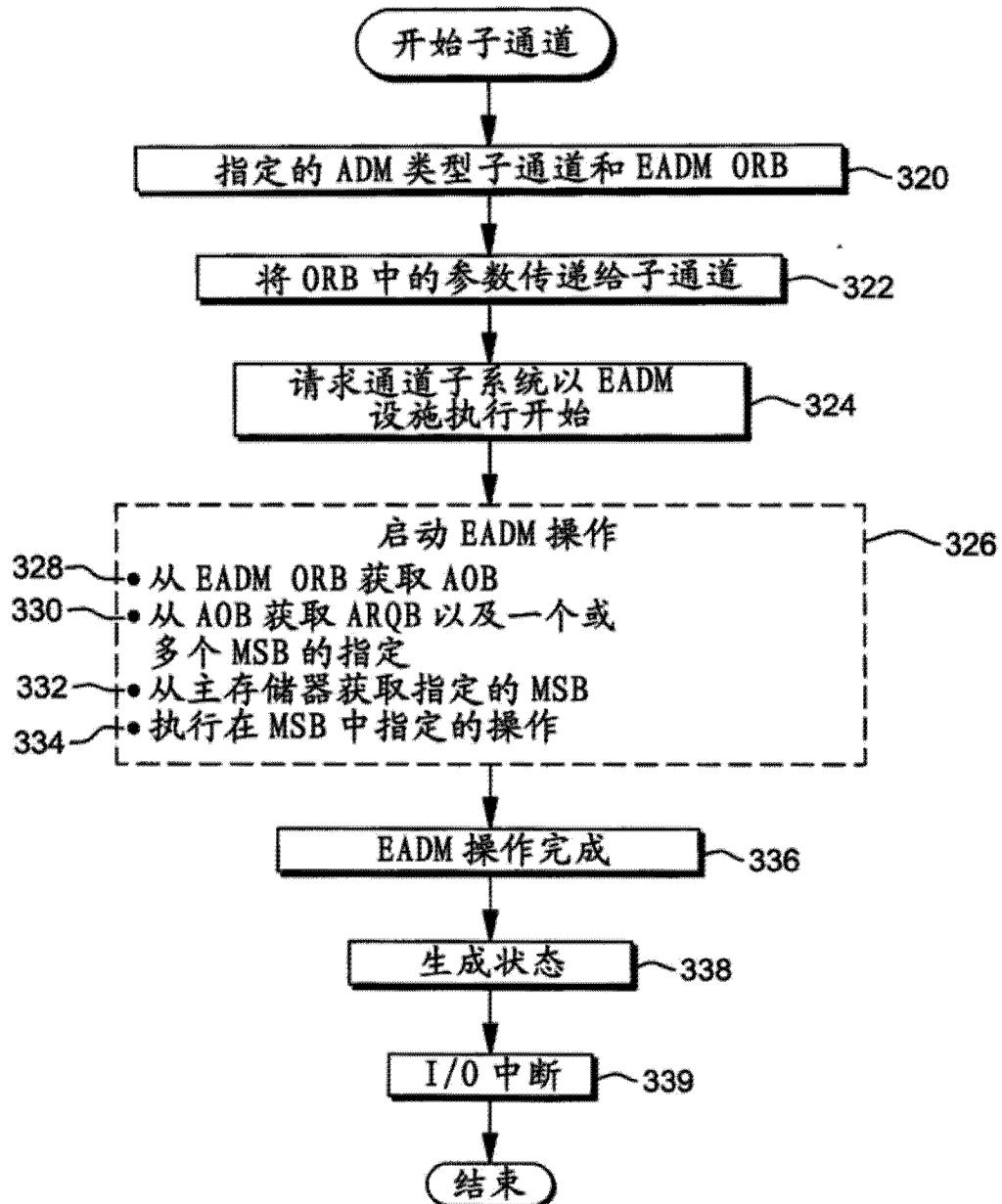


图 3B

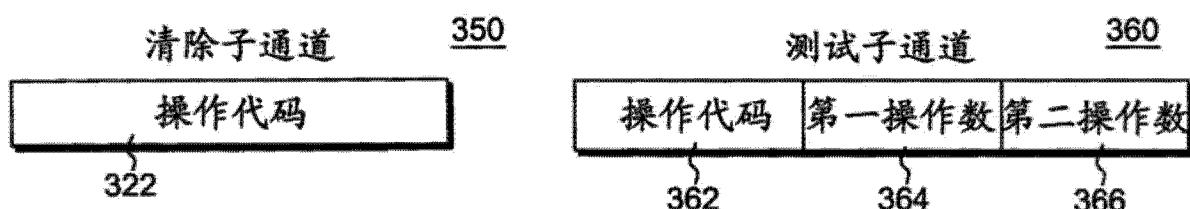


图 3C

图 3D

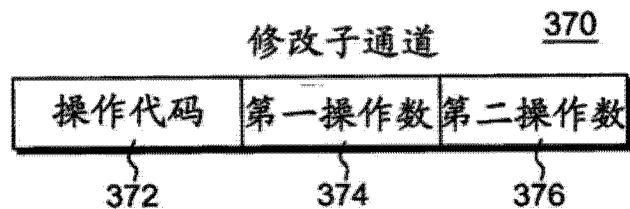


图 3E

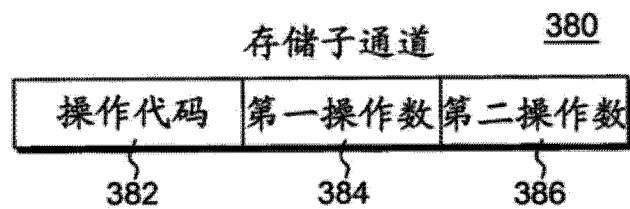


图 3F

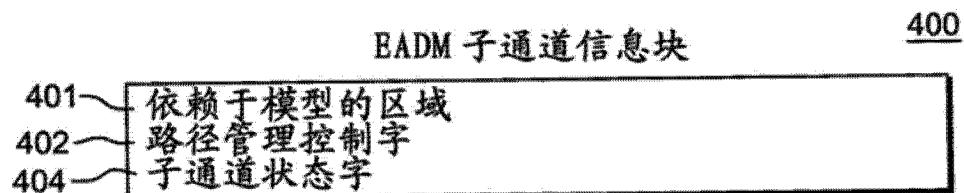


图 4A

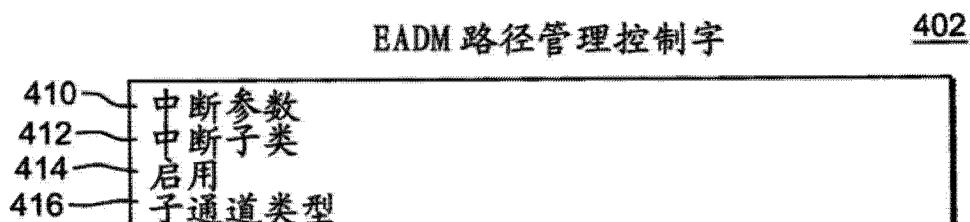


图 4B

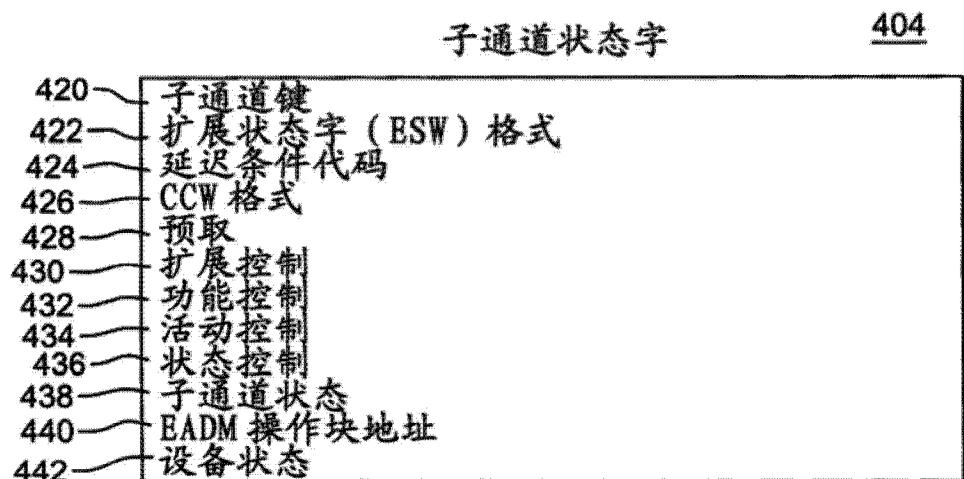


图 4C

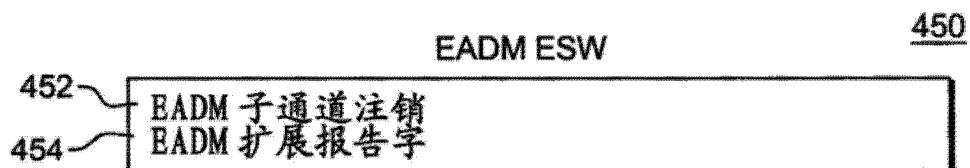


图 4D

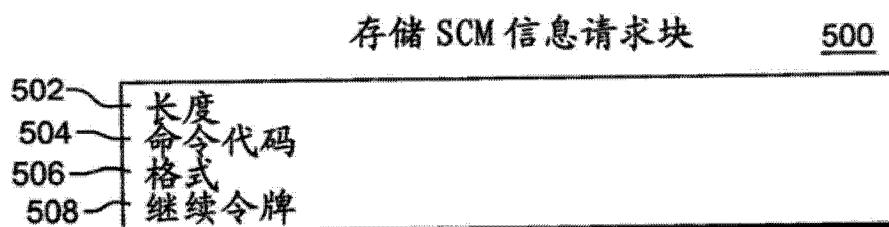


图 5A

存储 SCM 信息响应块

520

- | | |
|-----|---------------------|
| 522 | 长度 |
| 524 | 响应代码 |
| 526 | 格式 |
| 528 | RQ |
| 530 | 每 MSB 的最大块计数 |
| 532 | 最大 SCM 地址 |
| 534 | SCM 增量大小 |
| 536 | 最大 MSB 计数 |
| 538 | 最大可配置 SCM 增量 |
| 540 | CPC 的总的初始化的 SCM 增量 |
| 542 | CPC 的总的未初始化的 SCM 增量 |
| 544 | SCM 测量块大小 |
| 546 | SCM 资源部件的最大数量 |
| 548 | SCM 数据单元大小 |
| 550 | 继续令牌 |
| 552 | 存储级内存地址列表 |

图 5B

存储级内存地址列表条目

552

- | | |
|-----|--------|
| 560 | SCM 地址 |
| 562 | 持久性属性 |
| 564 | OP 状态 |
| 566 | 数据状态 |
| 568 | 等级 |
| 570 | R |
| 572 | 资源 ID |

图 5C

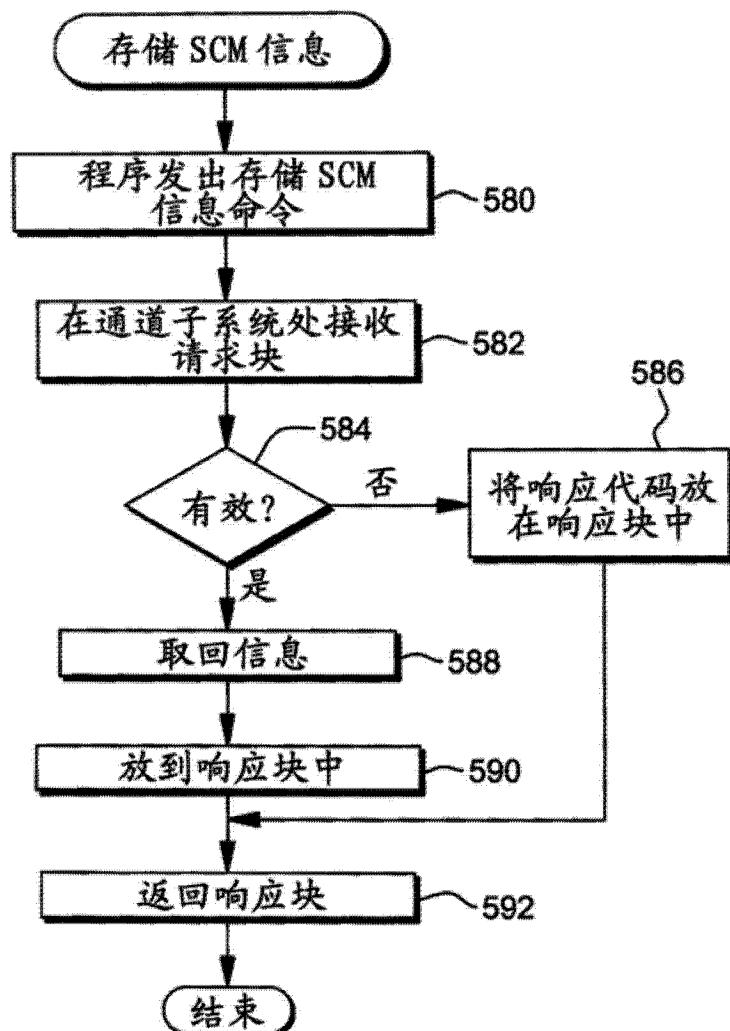


图 5D

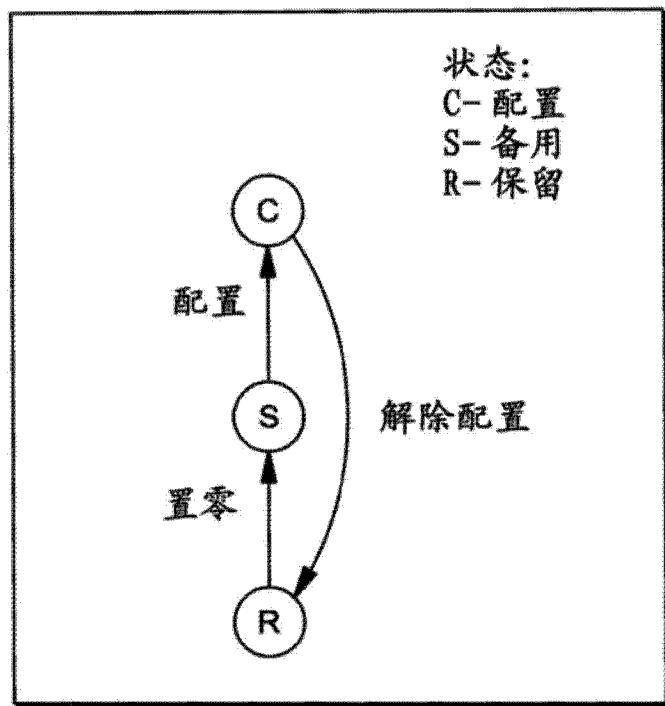


图 6A

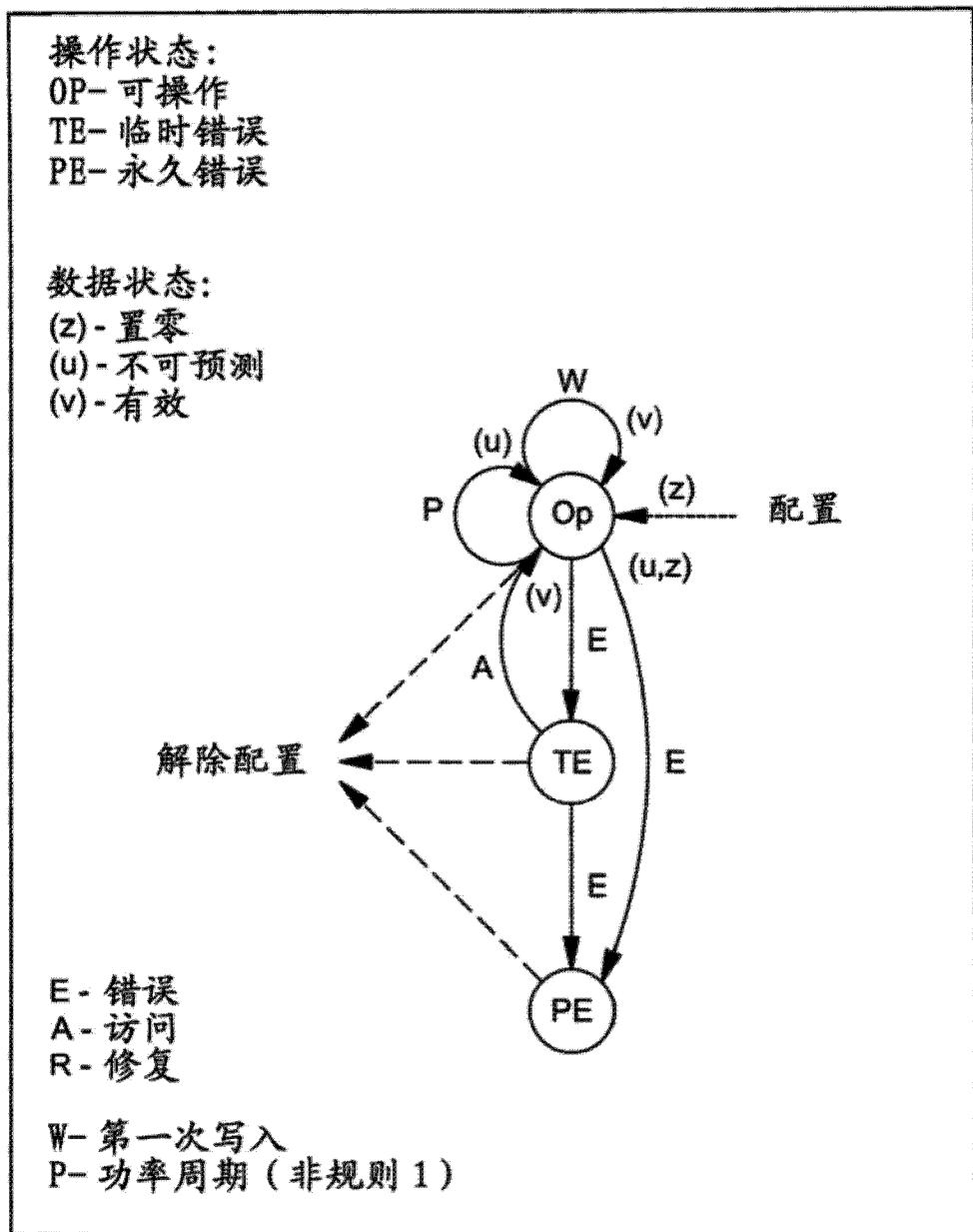


图 6B

配置存储级内存请求块

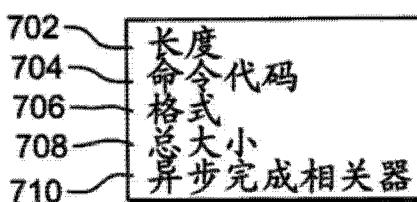
700

图 7A

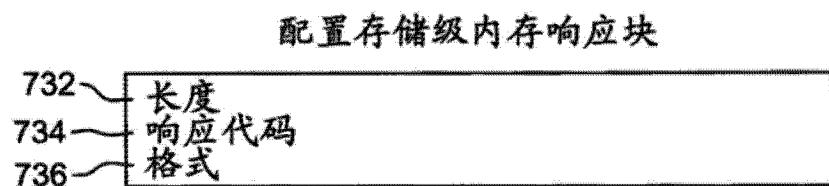
730

图 7B

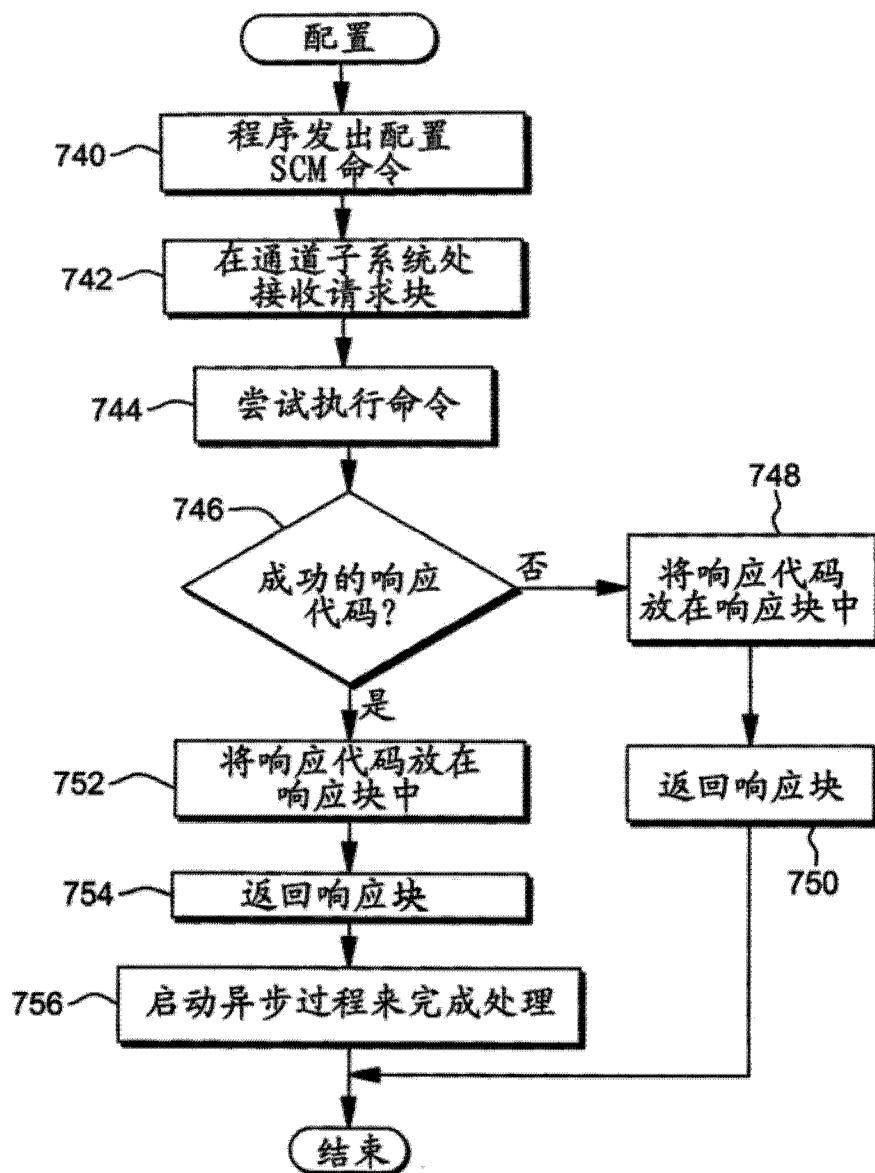


图 7C

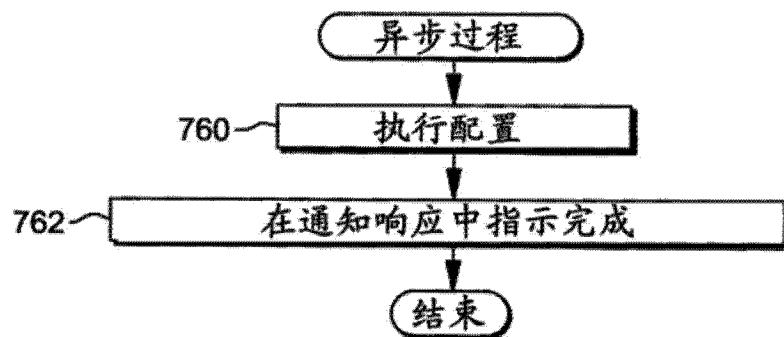


图 7D

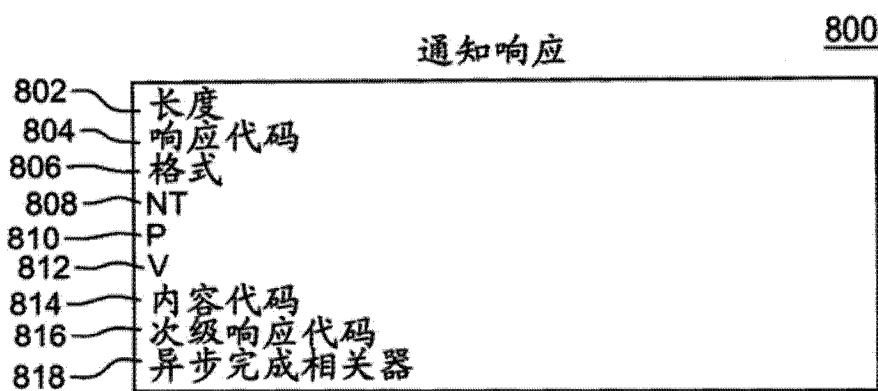


图 8A

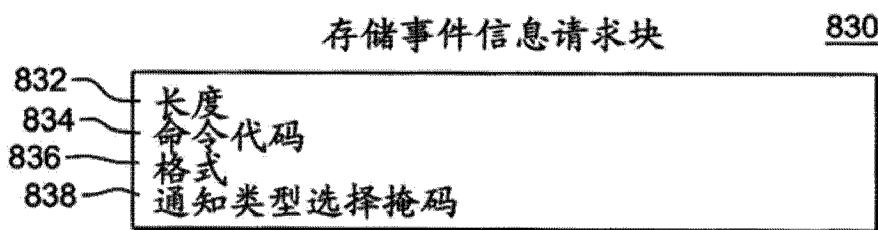


图 8B

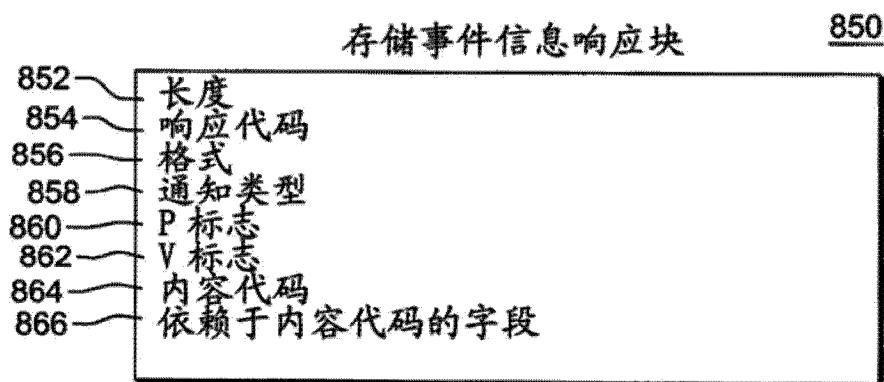


图 8C

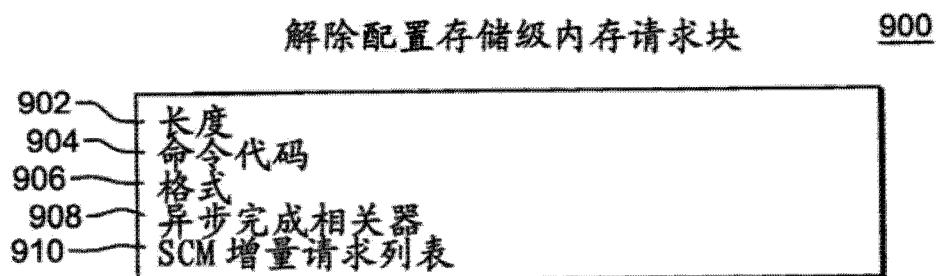


图 9A

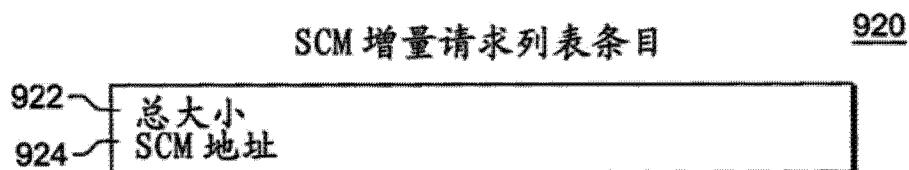


图 9B

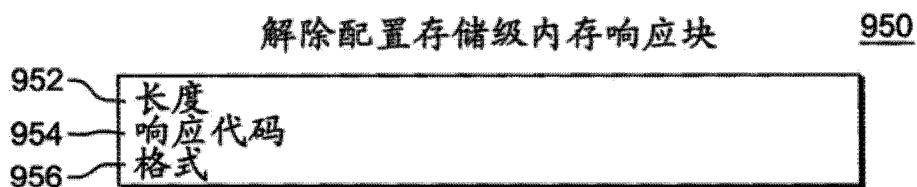


图 9C

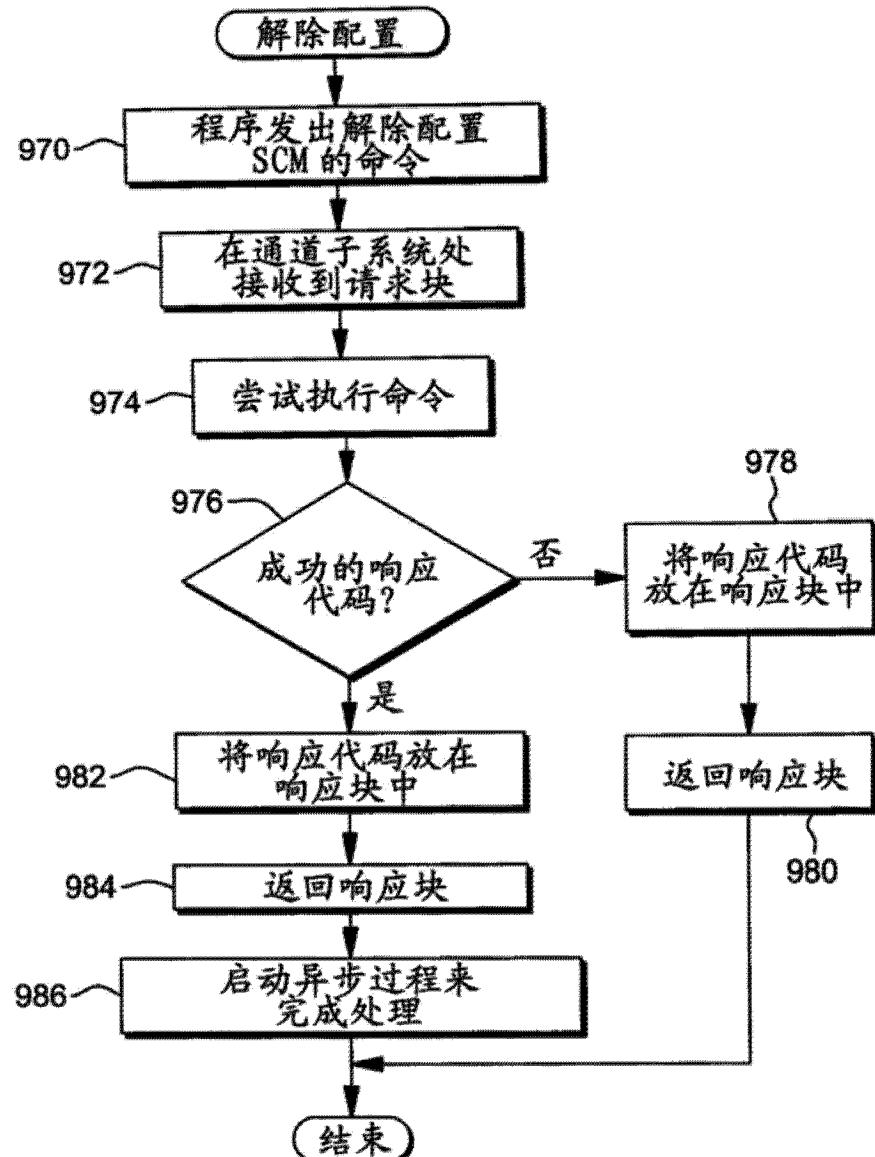


图 9D

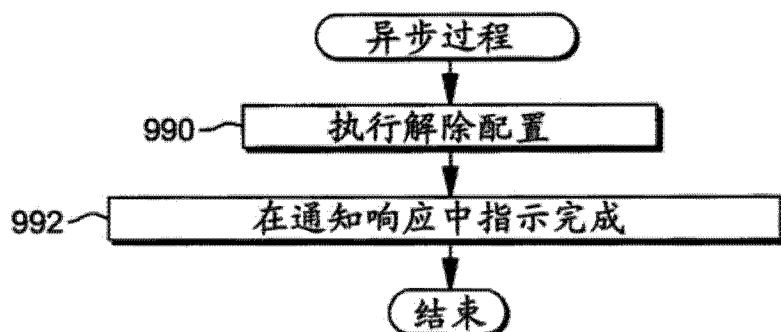


图 9E

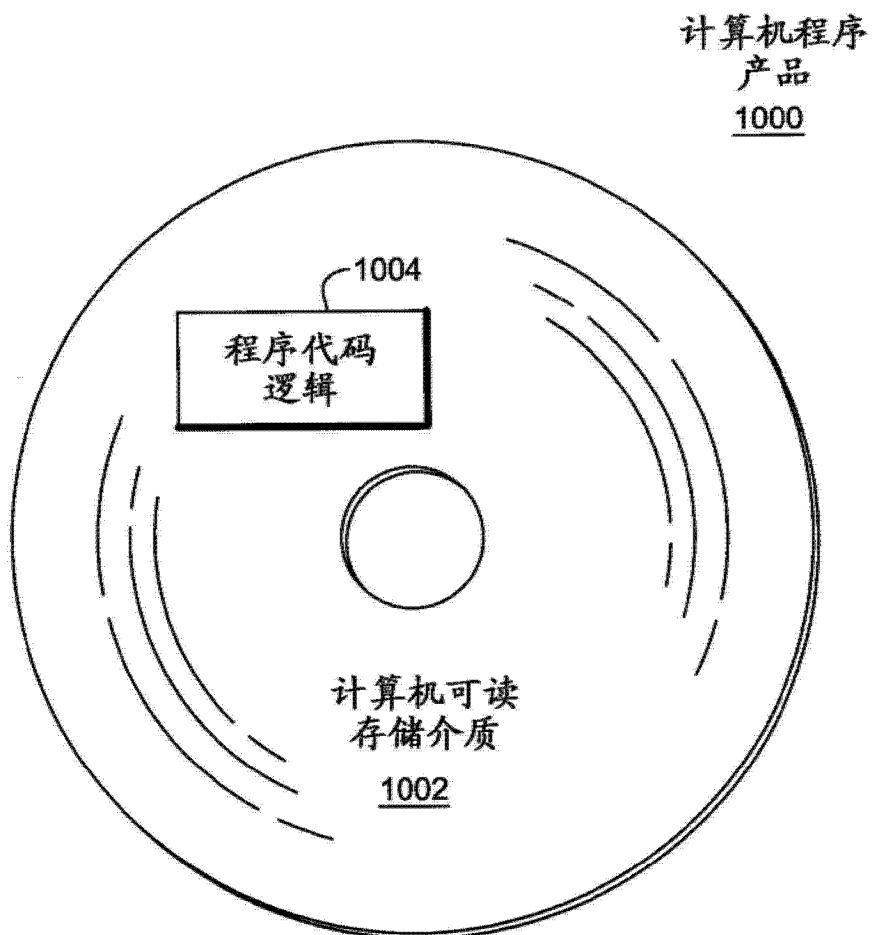


图 10

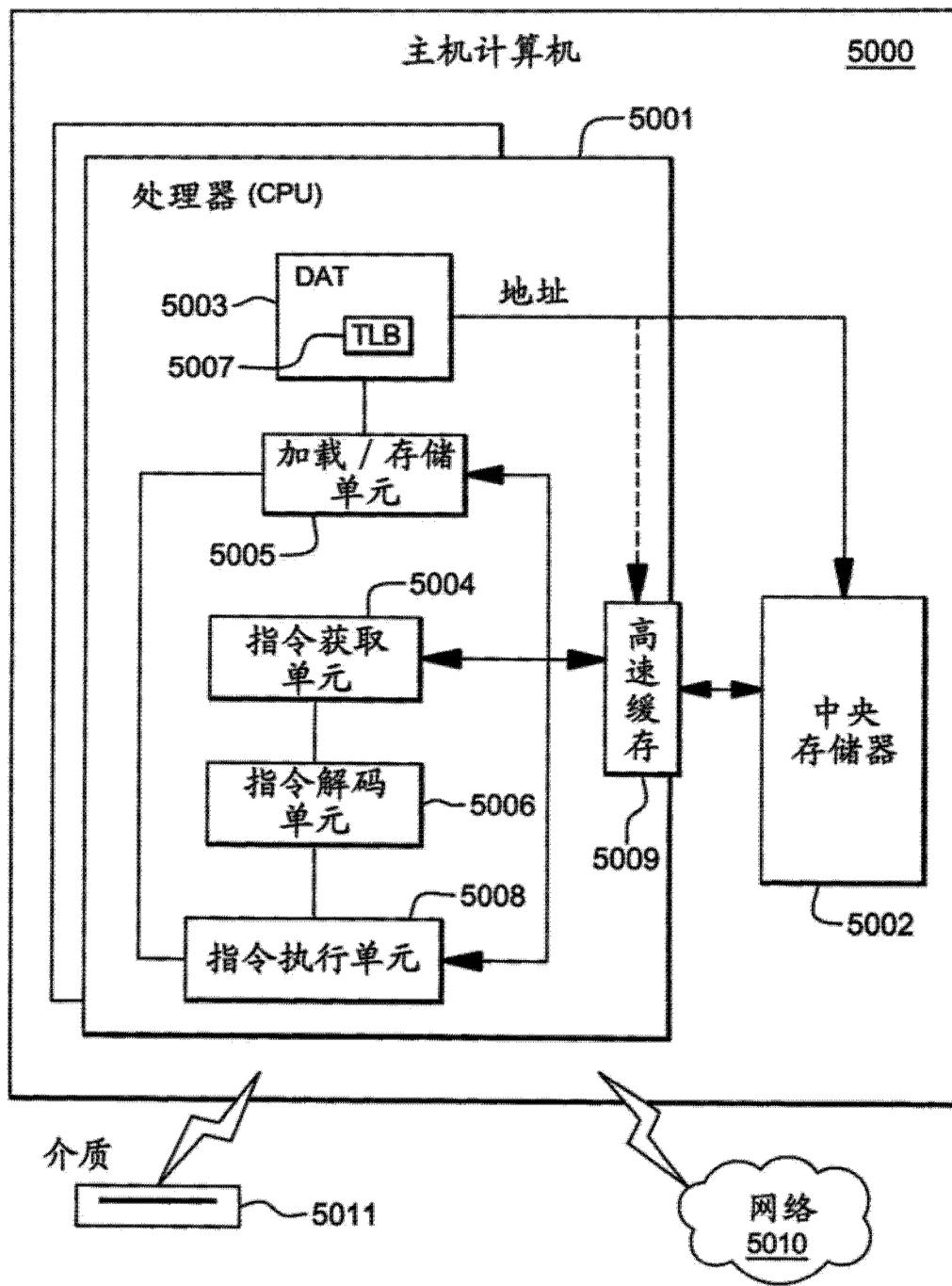


图 11

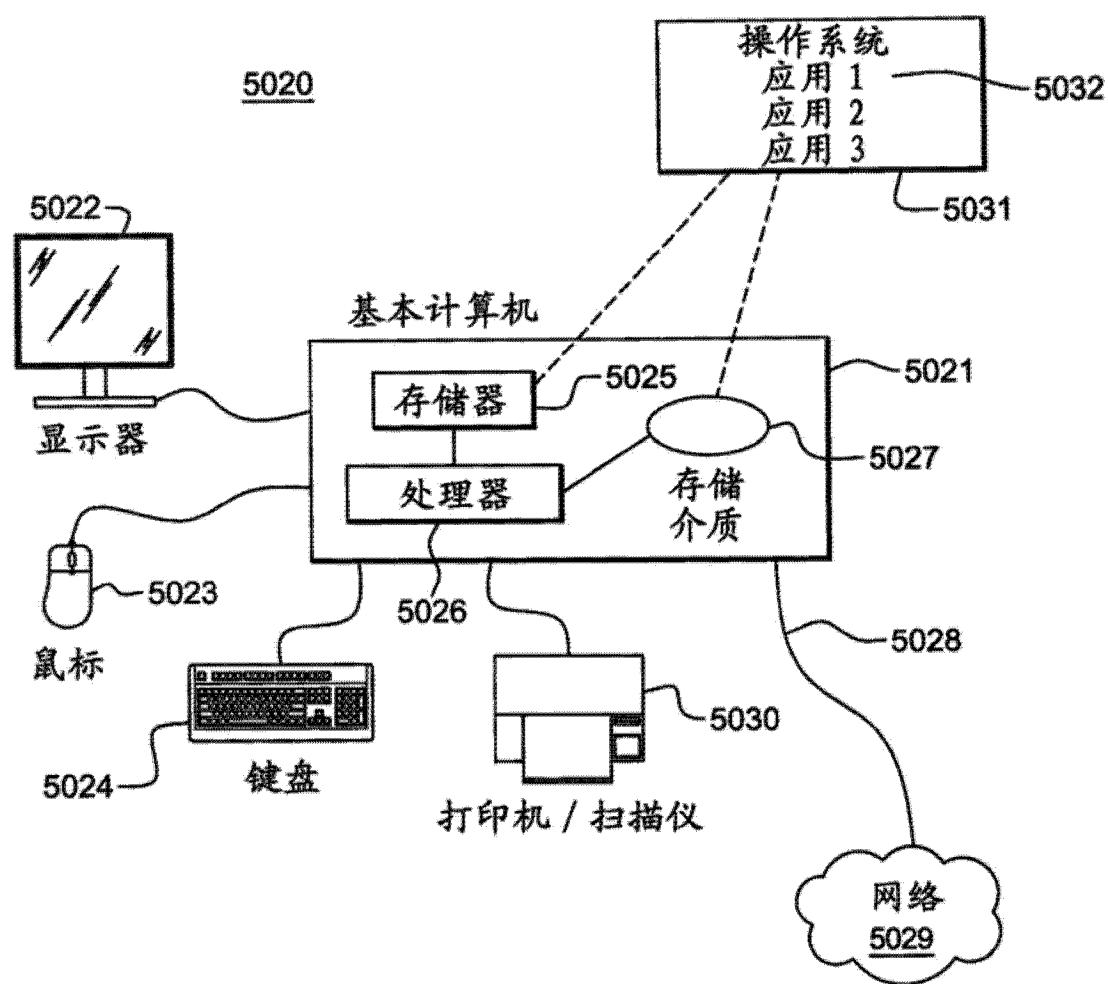


图 12

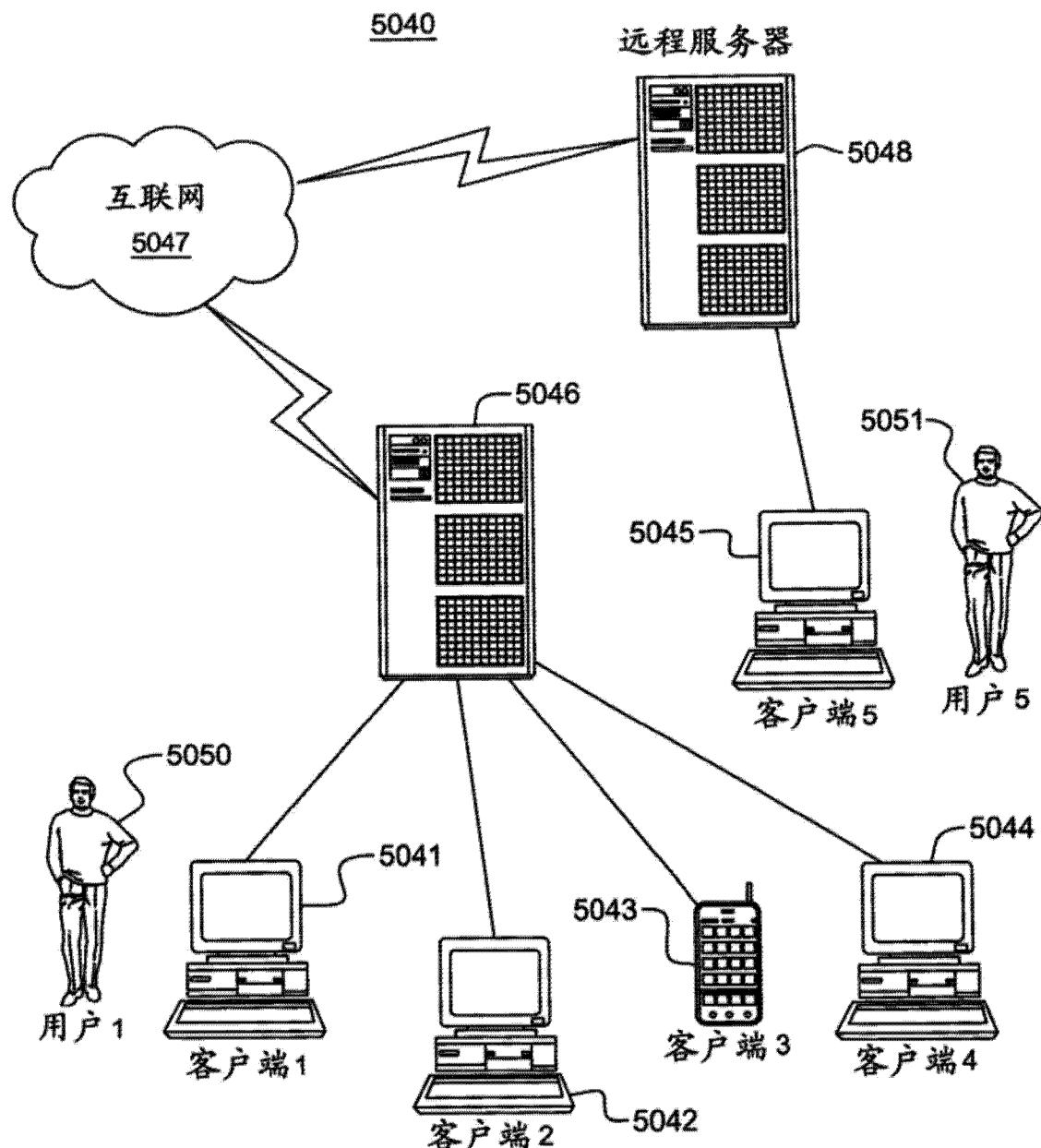


图 13

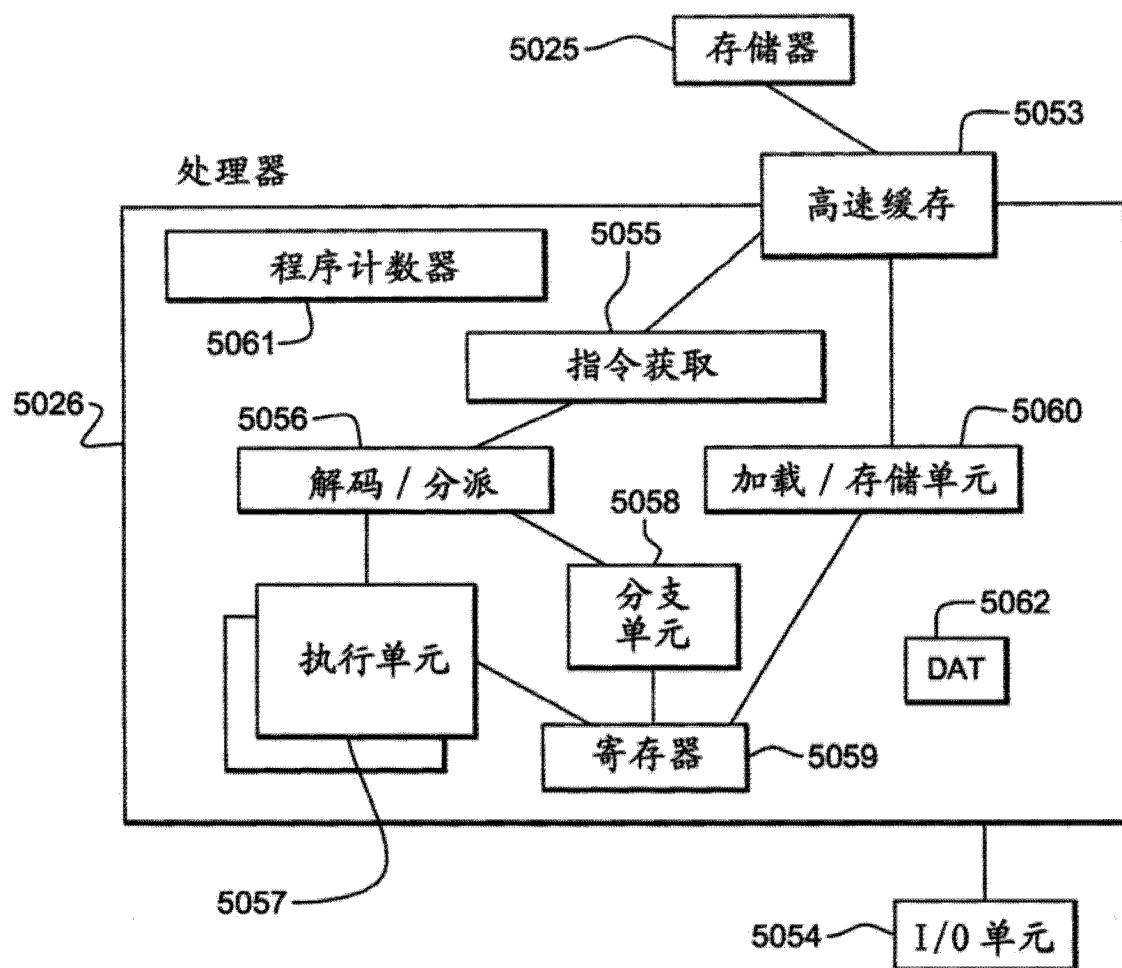


图 14

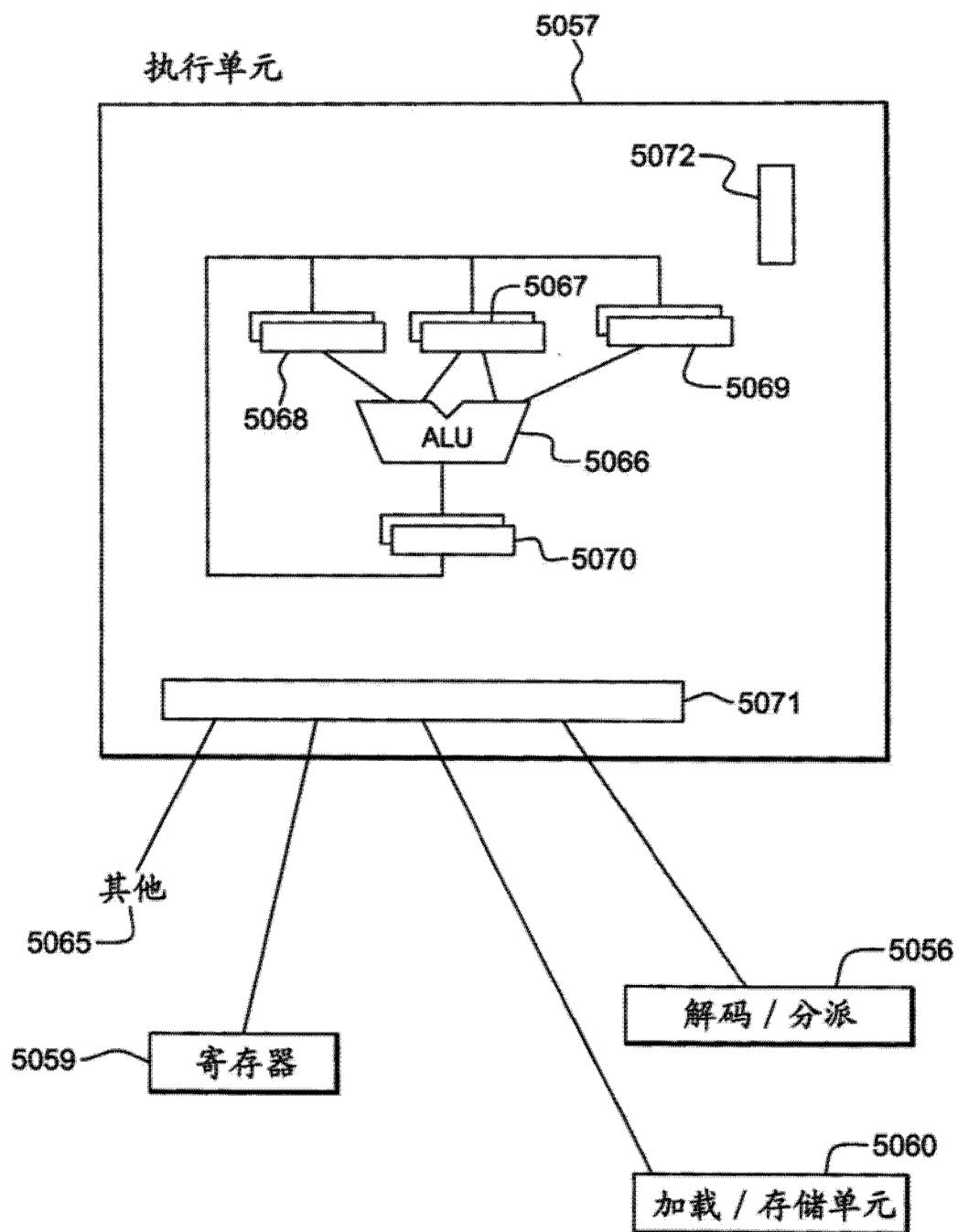


图 15A

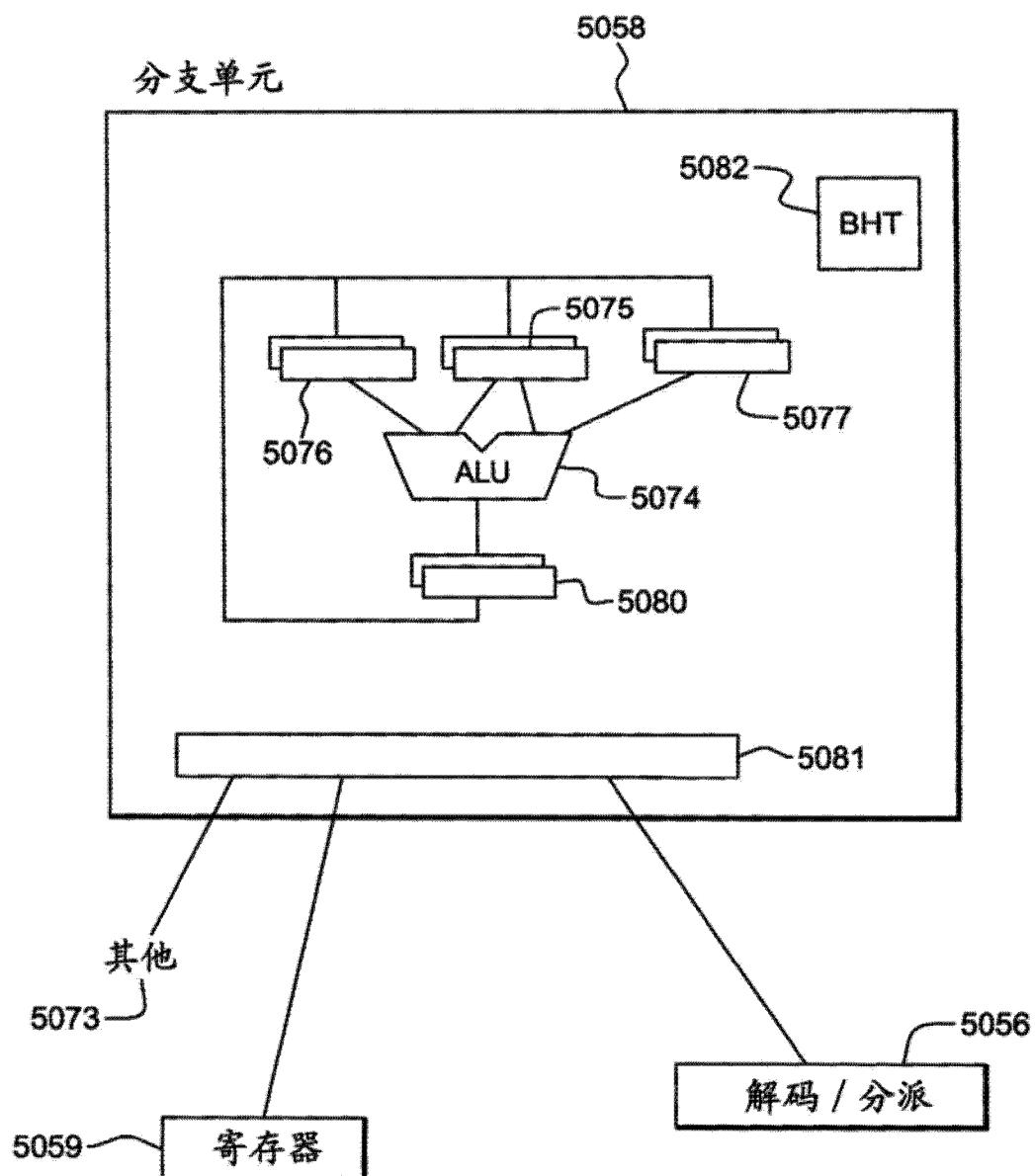


图 15B

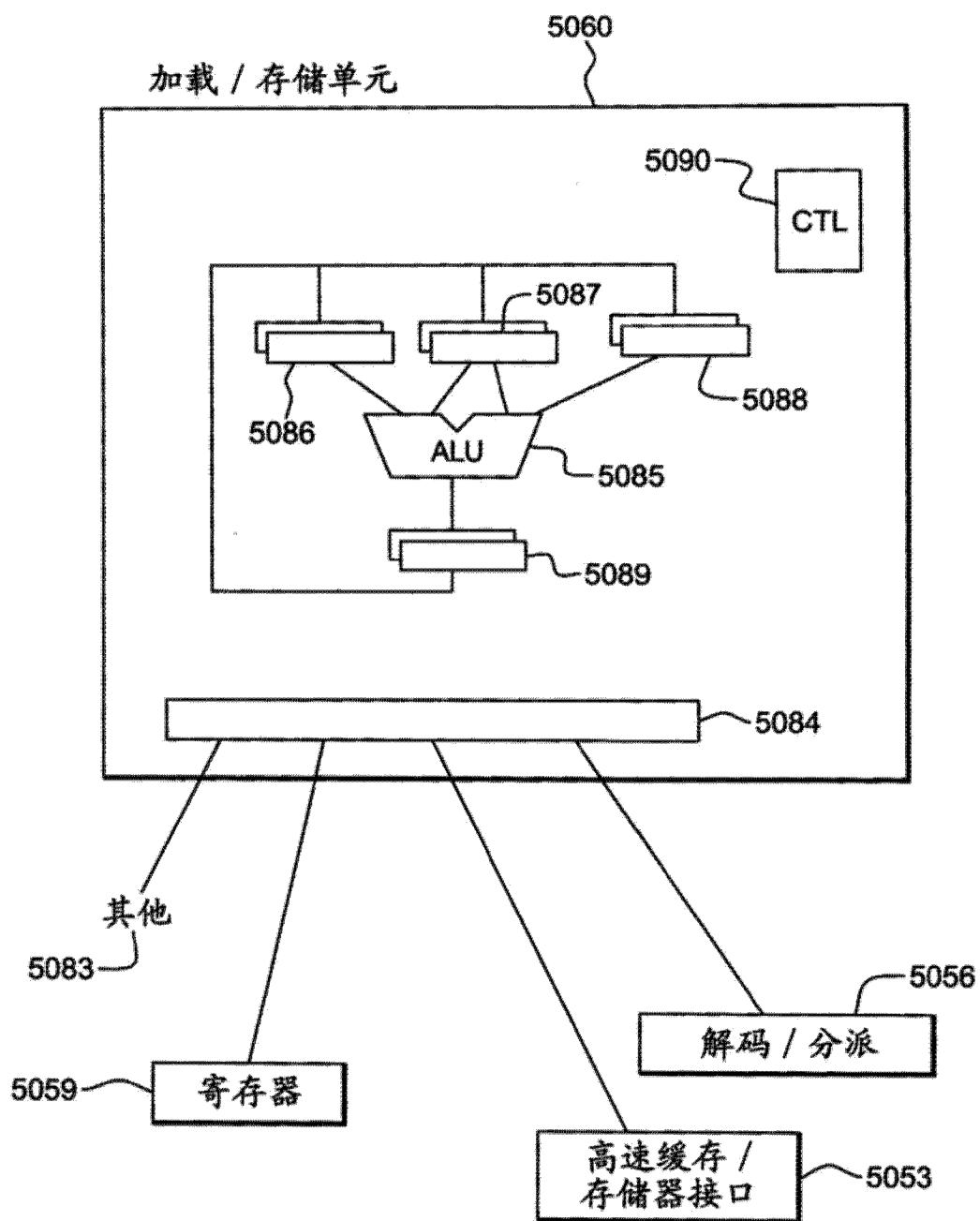


图 15C

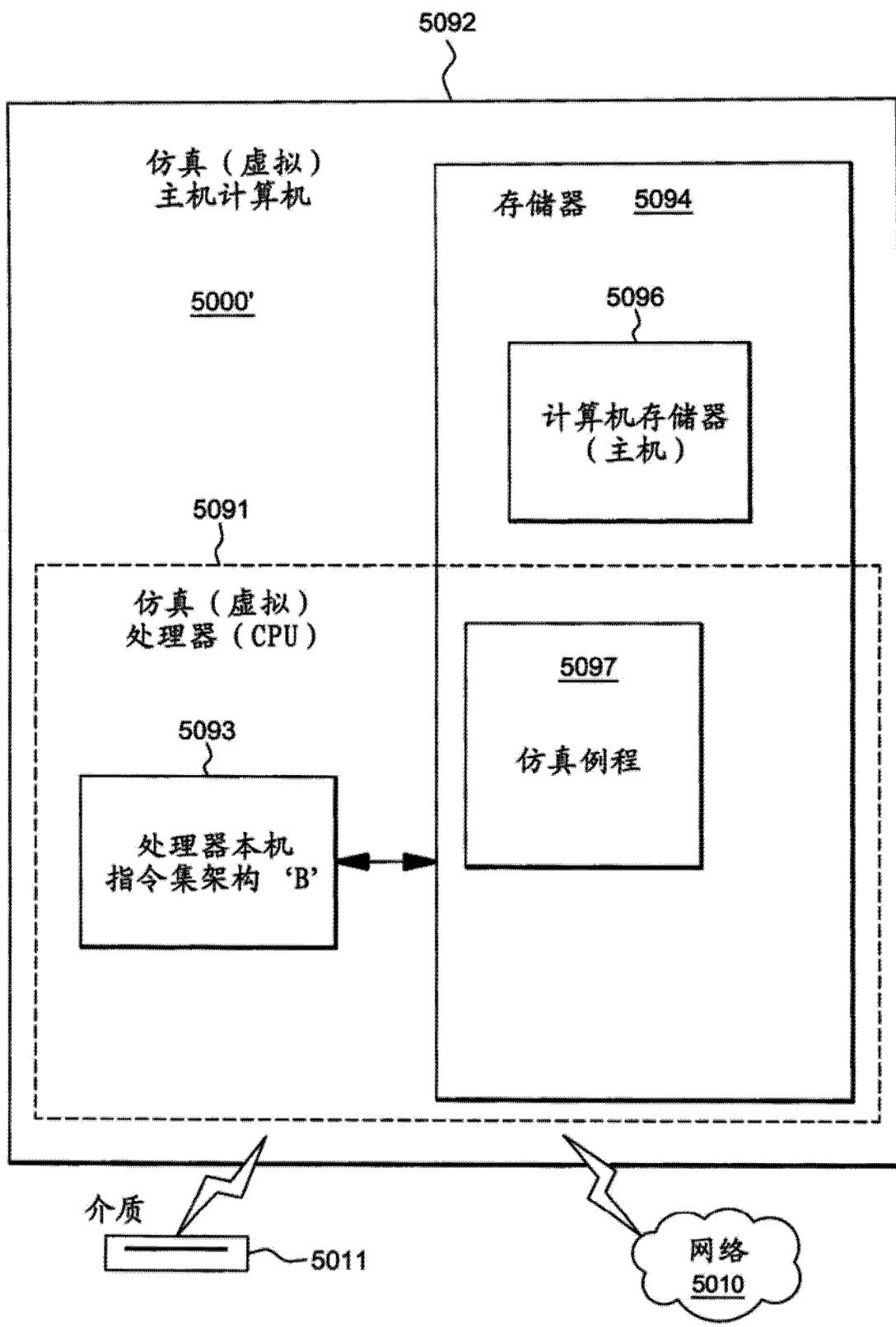


图 16