[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 3/06

[12] 发明专利申请公开说明书

[21] 申请号 200410061822.7

[43] 公开日 2004 年 12 月 29 日

[11] 公开号 CN 1558321A

[22] 申请日 2004.6.25
[21] 申请号 200410061822.7
[30] 优先权
[32] 2003. 7. 2 [33] US [31] 60/483,926
[71] 申请人 普安科技股份有限公司
地址 台湾台北县
[72] 发明人 刘宁一 李泽涵 施明文 王源辉 包崇华

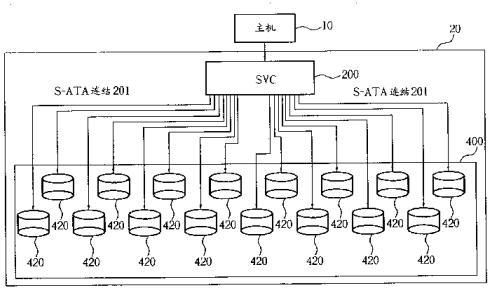
[74] 专利代理机构 北京市柳沈律师事务所
代理人 蒲迈文 黄小临

权利要求书 7 页 说明书 25 页 附图 20 页

[54] 发明名称 储存虚拟化计算机系统及用于其中的外接式控制器

[57] 摘要

本发明提供一种储存虚拟化计算机系统及用于其中的外接式控制器，该储存虚拟化计算机系统包含有一主机用来发出一输出入请求，一储存虚拟化控制器耦接于该主机、用来执行输出入操作以响应于该输出入请求，及至少一实体储存装置。该实体储存装置经由一点对点序列讯号连结耦接于该储存虚拟化控制器，用来通过该储存虚拟化控制器提供该主机储存空间。该点对点序列讯号连结可为一 SATA 输出入装置连结。



ISSN1008-4274

1. 一种储存虚拟化计算机系统包含有:

一主机，用来发出输出入请求;

一外接式储存虚拟化控制器，该储存虚拟化控制器耦接于该主机且用于执行输出入操作以响应于该输出入请求；以及

至少一实体储存装置，各实体储存装置经由一点对点序列讯号连结耦接于该储存虚拟化控制器，用来通过该储存虚拟化控制器提供该储存虚拟化计算机系统储存空间。

2. 如权利要求 1 所述的储存虚拟化计算机系统，其中该点对点序列讯号连结是指一序列先进技术接取接口 (SATA) 输出入讯号连结。

3. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该至少一实体储存装置包含有一序列先进技术接取接口 (SATA) 实体储存装置。

4. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该至少一实体储存装置包含有一平行先进技术接取接口 (PATA) 实体储存装置，同时，一序列至并行转换器耦接于该储存虚拟化控制器与该平行先进技术接取接口 (PATA) 实体储存装置。

5. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其还包含有一附加于该储存虚拟化控制器的可拆卸匣用以容置该至少一实体储存装置之一于其中。

6. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该至少一实体储存装置在该储存虚拟化控制器处于线上状况时可由该储存虚拟化控制器上拆卸下来或附加上去。

7. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该储存虚拟化控制器是构造成定义由该至少一实体储存装置的区段所组成的至少一逻辑介质单元。

8. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该储存虚拟化控制器包含有:

一中央处理电路，用于执行输出入操作以响应于该主机的该输出入请求;

至少一输出入装置连结控制器，耦接于该中央处理电路;

至少一主机端输出入装置连结端口，设置于该至少一输出入装置连结控制器之一中，用来耦接至该主机；以及

至少一装置端输出入装置连结端口，设置于该至少一输出入装置连结控制器之一中，用来耦接至该至少一实体储存装置之一。

9. 如权利要求 8 所述的储存虚拟化计算机系统，其中该主机端输出入装置连结端口中之一与该装置端输出入装置连结端口中之一是设置于同一个该输出入装置连结控制器中。

10. 如权利要求 8 所述的储存虚拟化计算机系统，其中该主机端输出入装置连结端口中之一与该装置端输出入装置连结端口中之一是设置于不同的该输出入装置连结控制器中。

11. 如权利要求 1 或 2 所述的储存虚拟化计算机系统，其中该储存虚拟化控制器包含有多个主机端输出入装置连结端口，且每一个该主机端输出入装置连结端口用于耦接至一主机端输出入装置连结。

12. 如权利要求 11 所述的储存虚拟化计算机系统，其中该储存虚拟化控制器设置成可在该主机端输出入装置连结端口中的至少两个上冗余地呈现一逻辑介质单元。

13. 如权利要求 8 所述的储存虚拟化计算机系统，其中该至少一主机端输出入装置连结端口为下列之一：在目标模式时支持点对点连结的光纤信道，在目标模式时支持专用回路连结的光纤信道，在目标模式时支持公用回路连结的光纤信道，操作于目标模式的平行小型计算机系统接口(平行 SCSI)，操作于目标模式时支持因特网小型计算机系统接口(iSCSI)协议的以太网络，操作于目标模式的序列附加小型计算机系统接口(SAS)，以及操作于目标模式时的序列先进技术接取接口(SATA)。

14. 一储存虚拟化子系统，用来提供一主机储存空间，该储存虚拟化子系统包含有：

一外接式储存虚拟化控制器，用来连接至该主机，且执行输出入操作以响应于由该主机发出的输出入请求；以及

至少一实体储存装置，且各实体储存装置经由一点对点序列讯号连结耦接于该储存虚拟化控制器，用来通过该储存虚拟化控制器提供该主机储存空间。

15. 如权利要求 14 所述的储存虚拟化子系统，其中该点对点序列讯号连

结为一序列先进技术接取接口(SATA)输出入装置连结。

16.如权利要求 14 或 15 所述的储存虚拟化子系统，其中该储存虚拟化控制器包含有:

一中央处理电路，用以执行输出入操作以响应于该主机的输出入请求;

至少一输出入装置连结控制器，耦接于该中央处理电路;

至少一主机端输出入装置连结端口，设置于该至少一输出入装置连结控制器之一中，用来耦接至该主机; 以及

至少一装置端输出入装置连结端口，设置于该至少一输出入装置连结控制器的之一中，用来耦接至该至少一实体储存装置之一。

17.如权利要求 16 所述的储存虚拟化子系统，其中该主机端输出入装置连结端口的其中之一与该装置端输出入装置连结端口的其中之一是设置于同一该输出入装置连结控制器中。

18.如权利要求 16 所述的储存虚拟化子系统，其中该主机端输出入装置连结端口的其中之一与该装置端输出入装置连结端口的其中之一是设置于不同的该输出入装置连结控制器中。

19.如权利要求 16 所述的储存虚拟化子系统，其中该至少一实体储存装置包含有一序列先进技术接取接口(SATA)实体储存装置。

20.如权利要求 16 所述的储存虚拟化子系统，其中该储存虚拟化控制器包含有多个主机端输出入装置连结端口，且每一个该主机端输出入装置连结端口用于耦接至一主机端输出入装置连结。

21.如权利要求 16 所述的储存虚拟化子系统，其中该储存虚拟化控制器构造成定义由该至少一实体储存装置的区段所组成的至少一逻辑介质单元。

22.如权利要求 20 所述的储存虚拟化子系统，其中该储存虚拟化控制器设置成在该主机端输出入装置连结端口中的至少两个上冗余地呈现一逻辑介质单元。

23.如权利要求 16 所述的储存虚拟化子系统，其中该至少一实体储存装置包含有一平行先进技术接取接口(PATA)实体储存装置，同时，一序列至并行转换器耦接于该装置端输出入装置连结端口与该平行先进技术接取接口(PATA)实体储存装置之间。

24.如权利要求 16 所述的储存虚拟化子系统，其还包含有一附加于该储存虚拟化控制器的可拆卸匣，用以容置该至少一实体储存装置之一于其中。

25. 如权利要求 16 所述的储存虚拟化子系统，其中该至少一实体储存装置在该储存虚拟化控制器处于线上状况时可由该储存虚拟化控制器上拆卸下来或附加上去。

26. 如权利要求 16 所述的储存虚拟化子系统，其中该储存虚拟化控制器还包含至少一多装置装置端扩充端口，该多装置装置端扩充端口用来支持一组额外的至少一实体储存装置。

27. 如权利要求 16 所述的储存虚拟化子系统，其中该至少一主机端输出入装置连结端口为下列之一：在目标模式时支持点对点连结的光纤信道，在目标模式时支持专用回路连结的光纤信道，在目标模式时支持公用回路连结的光纤信道，操作于目标模式的平行小型计算机系统接口(平行 SCSI)，操作于目标模式时支持因特网小型计算机系统接口(iSCSI)协议的以太网络，操作于目标模式的序列附加小型计算机系统接口(SAS)，以及操作于目标模式时的序列先进技术接取接口(SATA)。

28. 如权利要求 16 所述的储存虚拟化子系统，还包含有一箱体管理服务(EMS)机制。

29. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制管理并监视该储存虚拟化子系统中的至少一下列装置：电源供应器、风扇、温度感知器、电压、不断电系统、电池、发光二极管(LED)、声响警报器、实体储存装置匣锁以与门锁。

30. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制设置为下列的一组态：支持直接连结箱体管理服务组态，支持装置代传箱体管理服务组态，以及同时支持直接连结箱体管理服务与装置代传箱体管理服务组态。

31. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制设置成支持下列的一协议：小型计算机系统接口箱体服务(SES)的箱体管理服务协议，以及小型计算机系统接口存取容错箱体(SAF-TE)的箱体管理服务协议。

32. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制中用来与该储存虚拟化控制器通联者包含下列之一：集成电路间(I2C)锁存，状态监视电路，以及同时具有集成电路间(I2C)锁存与状态监视电路。

33. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制

还包含有一用来执行程序的中央处理器。

34. 如权利要求 28 所述的储存虚拟化子系统，其中该箱体管理服务机制还包含有至少一集成电路间(I2C)连结，用来作为连接至该储存虚拟化控制器的主要通讯媒介。

35. 一种外接式储存虚拟化控制器，用来执行输出入操作以响应于来自一主机的输出入请求，该外接储存虚拟化控制器包含有:

一中央处理电路，用以执行输出入操作以响应该主机的输出入请求;

至少一输出入装置连结控制器，耦接于该中央处理电路;

至少一主机端输出入装置连结端口，设置于该至少一输出入装置连结控制器之一中，用来耦接至该主机; 以及

至少一装置端输出入装置连结端口，设置于该至少一输出入装置连结控制器之一中，用来耦接至至少一实体储存装置并与的执行点对点序列讯号传递。

36. 如权利要求 35 所述的外接式储存虚拟化控制器，其中该装置端输出入装置连结控制器包含有至少一序列先进技术接取接口(SATA)端口，每一该序列先进技术接取接口(SATA)端口通过一序列先进技术接取接口(SATA)输出入装置连结与该至少一实体储存装置之一连接。

37. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其中该主机端输出入装置连结端口的其中之一及该装置端输出入装置连结端口的其中之一设置于同一个该输出入装置连结控制器中。

38. 如权利要求 35 或第 36 所述的外接储存虚拟化控制器，其中该主机端输出入装置连结端口的其中之一及该装置端输出入装置连结端口位的其中之一设置于不同的该输出入装置连结控制器中。

39. 如权利要求 35 或 36 所述的外接式储存虚拟化控制器，其中该装置端输出入装置连结控制器以下列的一接口而与该中央处理电路连结: 周边组件连结(PCI)接口，周边组件连结扩充(PCI-X)接口，以及周边组件连结快捷(PCI Express)接口。

40. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其中该外接式储存虚拟化控制器包括多个主机端输出入装置连结端口，各用于耦接于一主机端输出入装置连结。

41. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其中该储存虚

拟化控制器构造成定义由该至少一实体储存装置的区段所组成的至少一逻辑介质单元。

42. 如权利要求 40 所述的外接储存虚拟化控制器，其中该储存虚拟化控制器设置成在该主机端输出入装置连结端口中的至少两个上冗余地呈现一逻辑介质单元。

43. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其中该至少一实体储存装置为直接存取储存装置，及该储存虚拟化控制器构造成定义由该至少一直接存取储存装置所组成的至少一逻辑介质单元，且该逻辑介质单元依据磁盘阵列型态、或由磁盘阵列型态的组合来决定，藉此该主机可对该逻辑介质单元连续寻址。

44. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其还包含至少一多装置装置端扩充端口，该多装置装置端扩充端口用来支持一第二组至少一实体储存装置。

45. 如权利要求 35 或 36 所述的外接储存虚拟化控制器，其中该至少一主机端输出入装置连结端口为下列之一：在目标模式时支持点对点连结的光纤信道，在目标模式时支持专用回路连结的光纤信道，在目标模式时支持公用回路连结的光纤信道，操作于目标模式的平行小型计算机系统接口(平行 SCSI)，操作于目标模式时支持因特网小型计算机系统接口(iSCSI)协议的以太网络，操作于目标模式的序列附加小型计算机系统接口(SAS)，以及操作于目标模式时的序列先进技术接取接口(SATA)。

46. 如权利要求 35 或 36 所述的外接式储存虚拟化控制器，其还包含有一箱体管理服务机制。

47. 如权利要求 46 所述的外接式储存虚拟化控制器，其中该箱体管理服务机制设置成用来支持下列的一组态：支持直接连结箱体管理服务组态，支持装置代传箱体管理服务组态，以及同时支持直接连结箱体管理服务与装置代传箱体管理服务组态。

48. 如权利要求 46 所述的外接式储存虚拟化控制器，其中该箱体管理服务机制设置成用来支持下列之一协议：小型计算机系统接口箱体服务(SES)的箱体管理服务协议，以及小型计算机系统接口存取容错箱体(SAF-TE)的箱体管理服务协议。

49. 一种执行储存虚拟化于一具有一外接式储存虚拟化控制器的计算机

系统中的方法，该方法包含:

以该外接式储存虚拟化控制器自该计算机系统中的一主机端接收一输出入请求;

以该储存虚拟化控制器剖析该输出入请求，用以决定至少一输出入操作来执行以回应于该输出入请求；以及

以该储存虚拟化控制器执行该至少一输出入操作，并以点对点序列讯息传递方式存取该计算机系统的至少一实体储存装置。

50.如权利要求 49 所述的方法，其中该点对点序列讯号传递是以符合序列先进技术接取接口(SATA)协议的格式进行。

51.如权利要求 49 或 50 所述的方法，其还包含有一提供箱体管理服务机制的步骤。

52.如权利要求 51 所述的方法，其中该箱体管理服务机制被设置成支持下列的一组态时，执行该箱体管理服务机制的步骤：支持直接连结箱体管理服务组态，支持装置代传箱体管理服务组态，以及同时支持直接连结箱体管理服务与装置代传箱体管理服务组态。

53.如权利要求 51 所述的方法，其中该储存虚拟化控制器被设置成支持下列的一协议时，执行该箱体管理服务机制的步骤：小型计算机系统接口箱体服务(SES)的箱体管理服务协议，以及小型计算机系统接口存取容错箱体(SAF-TE)的箱体管理服务协议。

54.如权利要求 49 或 50 所述的方法，其中执行该至少一输出入操作的步骤包含有发出至少一装置端输出入请求至一装置端输出入装置连结控制器、及将该装置端输出入请求与伴随的输出入数据再格式化成至少一传输用的数据包。

55.如权利要求 54 所述的方法，其中该数据包包含有一用来指示该数据包起始端且位于前端的起始段(start segment)、一用来指示该数据包终结端且位于尾端的终结段(end segment)、一经由该装置端输出入装置连结传送且含有实际输出入信息的有效负载数据段(payload data segment)、以及一含有由该有效负载数据导出并用来检核传送后的有效负载数据正确性的检验码(check code)的检验数据段(check data segment)。

储存虚拟化计算机系统及用于其中的外接式控制器

技术领域

本发明涉及一种储存虚拟化计算机系统，特别是涉及一种使用点对点序列讯号连结做为主要装置端输出入装置连结的储存虚拟化计算机系统。

背景技术

所谓储存虚拟化（storage virtualization）是一种将实体储存空间虚拟化的技术，其是将实体储存装置(PSDs)的不同区段结合成可供一主机系统存取使用的逻辑储存体（logical storage entity)-在此称为「逻辑介质单元」（logical media units，LMU）。该技术主要用于磁盘阵列(RAID)储存虚拟化，经由此磁盘阵列的技术，可将较小实体储存装置结合成为容量较大、可容错、高效能的逻辑介质单元。

储存虚拟化控制器(storage virtualization controller，SVC)的主要目的是将实体储存介质的各区段的组合映像(map)形成一主机系统可见的逻辑介质单元。由该主机系统发出的输出入（IO）请求于接收之后会先被剖析并解译，且相关的操作及数据会被编译成实体储存装置的输出入请求。这个过程可以是间接地，例如运用快取、延迟(如：回写(write-back))、预期（anticipate）(先读(read-ahead))、群集(group)等操作来加强效能及其它的操作特性，因而一主机输出入请求并不一定是以一对一的方式直接对应于实体储存装置输出入请求。

外接式（或可称为独立式(stand-alone)）储存虚拟化控制器为一种经由输出入接口连接于主机系统的储存虚拟化控制器，且其可连接至位于主机系统外部的装置，一般而言，外接式的储存虚拟化控制器通常是独立于主机进行运作。

外接式(或独立式)直接存取磁盘阵列控制器（external direct-access RAID controller）是外接式储存虚拟化控制器的一个例子。磁盘阵列控制器是将一个或多个实体直接存取储存装置（direct access storage devices,

DASDs)的区段组合以构成逻辑介质单元，而它们的构成方式由所采用的特定磁盘阵列型态(RAID level)决定，其所构成的逻辑介质单元对于主机系统而言，为可连续寻址的，以使每一逻辑媒逻辑介质单元可被利用。典型地，一个单一的磁盘阵列控制器 (single RAID controller)可支持多种磁盘阵列型态，因此，不同的逻辑介质单元可以由直接存取储存装置的各个区段藉由不同的磁盘阵列型态而以不同的方式组合而成，所组合成的各个不同的逻辑介质单元则具有各该磁盘阵列型态的特性。

另一个外接式储存虚拟化控制器的例子是 JBOD (Just a Bunch of Drives) 模拟控制器。JBOD 为『仅是一捆盘机』的缩写，是一组实体直接存取储存装置，并经由一个或多个多装置输出入装置连结信道 (multiple-device IO device interconnect channel) 直接连接于一主机系统上。但使用点对点输出入装置连结连接至该主机系统的直接存取储存装置 (如 SATA 硬盘、PATA 硬盘等)，无法通过直接连结而构成如前述的 JBOD 系统，因为这些直接存取储存装置并不允许多个装置直接连接至输出入装置信道。至于智能型的 JBOD 仿真器，是藉由将输出入请求映像到实体直接存取储存装置的方式，而用来仿真多个多装置输出入装置连结直接存取储存装置，而其中该实体直接存取储存装置是个别地经由点对点输出入装置连结信道连接至 JBOD 仿真器。

另一个外接式储存虚拟化控制器(缩写为 SVC)的例子为一种外接式磁带备份子系统。

储存虚拟化控制器最主要的功能是管理、结合及操控实体储存装置，并将其以一组逻辑介质单元的形式呈现于主机端，使各个逻辑介质单元在主机端看来，都像是直接连接至一个实体储存装置，而该逻辑介质单元则是该实体储存装置在逻辑上的等效物。为了要实现这个目的，由主机输出且由储存虚拟化控制器处理的输出入请求，若在一等效实体储存装置中通常会产生某些行为，则这些输出入请求会在储存虚拟化控制器关于所寻址的逻辑介质单元的部份上产生逻辑上等效的行为。其结果是，该主机会认为它是直接连接至一实体储存装置且与之通讯，虽然实际上，该主机连接至一仅是仿真该实体储存装置行为的储存虚拟化控制器上，而该 SVC 所寻址的逻辑介质单元乃该 PSD 的逻辑上的等效物。

为了要实现上述的行为模拟，储存虚拟化控制器将自主机接收来的输出入请求映像至逻辑上相等的内部操作，其中有部份的操作不需要产生任何装置端输出入请求至装置端实体储存装置便可以做完；这些操作仅需要在内部进行即可，并不需要对装置端实体储存装置进行存取。这类的输出入请求所产生的操作在此将称为「内部模拟操作(internally emulated operation)」。

然而，有些操作是无法单单经由内部模拟而执行的，但也无法直接对装置端实体储存装置进行存取。举例来说，如快取操作的数据读取操作时，对应于输出入请求所寻址的介质区段(media section)的数据目前刚好完全存在于储存虚拟化控制器的数据高速缓存中；或是在数据写入操作时，当该储存虚拟化控制器的高速缓存是操作于回写模式，则使数据先写入高速缓存中，而后才传送至适当的实体储存装置。这些操作都可称为「异步装置操作(asynchronous device operation)」，亦即为了使所请求的操作发生以实现其原来目的而传至装置端实体储存装置的所有实际的输出入请求都是间接地于所请求的操作之前或之后进行，而不是直接地响应于所请求的操作而进行。

另外还有一类由直接产生装置端输出入请求至实体储存装置的操作所构成的操作，这种操作一般可称做「同步装置操作(synchronous device operation)」。

此外，有一些主机端输出入请求可以映射至由多个不同类的子操作所组成的组合操作，这些子操作可以包括内部仿真操作、异步装置操作和/或同步装置操作。一个映像至异步装置操作及同步装置操作组合的主机端输出入请求的例子是，一个数据读取请求，其在逻辑介质单元中所寻址的介质区段所对应的数据，目前一部份存在于高速缓存当中，一部份不存在于高速缓存当中，因而必须从实体储存装置当中读取。这些从高速缓存当中读取数据的子操作是异步装置操作，因为这种子操作并不需要直接从装置端实体储存装置存取来做完此输出入请求，但是却间接依赖先前所执行的装置端实体储存装置存取的结果；而直接至实体储存装置读取数据的子操作则为同步装置操作，因为它所需要的是直接且立即的对装置端实体储存装置进行数据存取来做完此输出入请求。

传统上，一般储存虚拟化都是由平行SCSI(小型计算机系统接口，Small Computer System Interface)、光纤、或是PATA(平行先进技术接取接口，

Parallel Advanced Technology Attachment)输出入装置连结做为主要装置端输出入装置连结(primary device-side IO device interconnect)，以将实体储存装置连接到储存虚拟化控制器。平行 SCSI 及光纤皆为多装置输出入装置连结，而多装置输出入装置连结的频宽需由与其连接的所有主机及所有装置共享的。

请参考图 1,图 1 为使用平行SCSI 做为主要装置端输出入装置连结的传统储存虚拟化计算机系统的方块示意图。每个平行 SCSI 装置端输出入装置连结的总频宽上限为 320 MB/s，而如图 1 当中的四个平行SCSI 装置端连结的应用，它的累加频宽则为 1280 MB/s。请参阅图 2，图 2 为使用光纤信道仲裁循环(Fibre Channel Arbitrated Loop; FC-AL)为主要装置端输出入装置连结的传统储存虚拟化计算机系统的方块示意图。每个光纤 FC-AL 装置端输出入装置连结的总频宽限制为 200 MB/s，而如图 2 当中的四个平行光纤 FC-AL 装置端连结的应用，它的频宽则为 800 MB/s。

多装置端输出入装置连结，例如平行 SCSI，有如下的缺点—假如有一个坏掉的装置连接在多装置连结上时，其可能会干扰主机及其它共享连结的装置的通联和/或数据传输。而光纤 FC-AL 在实际应用的时候可以减低上述的顾虑至某一程度，因为它提供双轨冗余连结，这种双轨冗余连结为每个装置提供两条信道，以防一条通道断掉或是被阻断。然而，这样的设计依然较差于每一个储存装置有其专用的连结，这是因为，两条连结上各自独立的失效仍旧会造成两条连结同时无法作用的问题。然而，另一方面，若使用专用的连结，则可以确保连结间的讯号完整性(signal integrity)具有完全的独立性，此时其中一个装置损毁并不会影响其它装置。

另一个传统的储存虚拟化是使用 PATA 装置端输出入装置连结，这是一种使用平行讯号传输的点对点输出入装置连结。藉由使用此种点对点连结，每个实体储存装置都有各自的专用连结连接至主机端，每个各别的储存装置都有一个专用频宽,使N个实体储存装置可以实现单一连结通道N倍的频宽。

PATA 也有如下的缺点—此种输出入装置连结仅能保护信息的有效负载数据的部份，而非控制信息的部份 (如区块地址及数据长度等)。而且，因为形成每一个 PATA 连结需要使用的专用的讯号线的数目很多(为 28 个)，PATA 连结的数目在超过某一点后就不易增加。再者，由于 PATA 的平行特性，它无法支持更高的接口速度。

发明内容

因此本发明的目的之一，在于提供一种储存虚拟化计算机系统，这种储存虚拟化计算机系统使用点对点序列讯号传输做为主要装置端输出入装置连结，以期解决上述问题。

本发明提供一种储存虚拟化计算机系统，包含有一主机，用来发出输出入请求，一外接式储存虚拟化控制器，耦接至该主机，用以执行输出入操作以响应此输出入请求，及至少一实体储存装置，以点对点序列讯息连结耦接于储存虚拟化控制器，使能通过储存虚拟化控制器来对主机提供储存空间。在本发明中，SATA(序列先进技术接取接口，Serial Advanced Technology Attachment)输出入装置连结为一点对点序列讯号连结的一实施例。

本发明的优点之一是，在所提供的储存虚拟化计算机系统中使用 SATA 为主要装置端输出入装置连结，每个实体储存装置都有其专用连结至该储存虚拟化控制器。

本发明另一优点是该 SATA 输出入装置连结不仅保护信息的有效负载数据的部份，尚可保护控制信息。

再者，本发明还提供一储存虚拟化子系统，其包含一储存虚拟化控制器，用来连接至一主机，且执行输出入操作以响应于由此主机发出的输出入请求，以及至少一实体储存装置，是经由点对点序列讯号连结耦接于储存虚拟化控制器，使能通过储存虚拟化控制器提供主机储存空间。

本发明提供一外接式储存虚拟化控制器，包含有一中央处理电路，执行输出入操作以响应于一主机端所发出的输出入请求；至少一输出入装置连结控制器，耦接于中央处理电路；至少一主机端输出入装置连结端口，设置于至少一输出入装置连结控制器之一中，用来耦接至主机；以及至少一装置端输出入装置连结端口，设置于至少一输出入连结控制器之一中，用来耦接至至少一实体储存装置并执行点对点序列讯号传递。

本发明还提供一种执行储存虚拟化于一具有一外接式储存虚拟化控制器的计算机系统中的方法，该方法包含以下的步骤：以此外接式储存虚拟化控制器自此计算机系统中的一主机端接收一输出入请求的步骤；以此储存虚拟化控制器剖析输出入请求，用以决定至少一输出入操作来执行以回应于输出入请求的步骤；以及，以此储存虚拟化控制器执行此至少一输出入操作，

并以点对点序列讯息传递方式存取计算机系统的至少一实体储存装置的步骤。

前述本发明所提供的储存虚拟化的方法，其执行过程可以藉由软件程序完成，因此本发明可以以计算机语言撰写程序后再加载一计算机可读取记录介质中，该记录介质可以是 IC 芯片、硬盘、光盘或其它可记录软件程序的物品。

附图说明

图 1 为使用平行 SCSI 作主要装置端输出入连结的传统储存虚拟化计算机系统方块图。

图 2 为使用平行光纤 FC-AL 作主要装置端输出入连结的传统储存虚拟化计算机系统方块图。

图 3 为本发明中储存虚拟化计算机系统的一实施例方块图。

图 4 为图 3 中储存虚拟化控制器及其连接至主机与实体储存装置阵列的实施例方块图。

图 5 为图 4 中中央处理电路的实施例方块图。

图 6 为图 5 中 CPU 芯片组/奇偶校验引擎的实施例方块图。

图 7 为图 4 中 SATA 装置连结控制器的方块图方块图。

图 8 为图 7 中的 PCI-X 至 SATA 控制器的实施例方块图。

图 9 为图 8 中 SATA 端口的方块图。

图 10 为例示一符合 SATA 协议的传输结构。

图 11 为例示一符合 SATA 协议的第一 FIS 数据结构。

图 12 为例示一符合 SATA 协议的第二 FIS 数据结构。

图 13 及 14 为图 3 中主机及储存虚拟化控制器间的输出入流程实例。

图 15 及 16 为图 3 中储存虚拟化控制器及实体储存装置间的输出入流程实例。

图 17 为支持装置端扩充端口的储存虚拟化子系统的方块图。

图 18 为另一支持装置端扩充端口的储存虚拟化子系统的方块图。

图 19 为可拆卸 PATA 实体储存装置匣的方块图。

图 20 为可拆卸 SATA 实体储存装置匣的方块图。

附图符号说明

主机	10	储存虚拟化子系统	20
储存虚拟化控制器	200	SATA 连结	201
主机端输出入装置连结控制器	220	存储器	280
中央处理电路	240	箱体管理服务电路	360
CPU	242	ROM	246
NVRAM	248	LCD	350
CPU 芯片组/奇偶校验引擎	224	CPU 接口	910
存储器接口	920	CM 先进先出缓冲器	922
除错码产生电路	924	除错码修正电路	926
PCI 接口	930, 932	PM 先进先出缓冲器	934, 936
X-BUS 接口	940	PM BUS	950
奇偶校验引擎	260	XOR 引擎	262
XOR 先进先出缓冲器	264	锁相回路	980
计时控制器	982	内部缓存器	984
UART 功能方块	986	SATA 输出入装置连结控制器	300
PCI 至 SATA 控制器	310	PCI-X 接口	312
组态电路	316	BUS 接口	318
Dec/Mux 仲裁器	314	SATA 端口	600
直接存储器存取缓存器	620	超集缓存器	630
指令区块缓存器	640	控制区块缓存器	650
双端口先进先出缓冲器	660	直接存储器存取控制器	670
PIO	680	传输层	690
连结层	700	实体层	710
实体储存装置阵列	400	实体储存装置	420

具体实施方式

请参考图 3，图 3 为本发明中储存虚拟化计算机系统的一实施例方块示意图，其主要装置端输出入装置连结为 SATA。该计算机系统包含有一主机 10 及一连接其上的储存虚拟化子系统(SVS)20。虽于此实施例当中仅有一主机 10 与一储存虚拟化子系统 20 相互连接，实际应用时可用一主机 10 连接多个储存虚拟化子系统 20,或是多个主机 10 连接一个储存虚拟化子系统 20，或是多主机 10 连接多个储存虚拟化子系统 20。

主机 10 可为一主机计算机，如一服务器系统、工作站、个人计算机系统等，而该储存虚拟化子系统包含有一储存虚拟化控制器(SVC) 200，此储存虚拟化控制器 200 可为一磁盘阵列控制器或是一 JBOD 仿真器，以及一利用 SATA 连结 201 连接至储存虚拟化控制器 200 的实体储存装置阵列 400 (physical storage device array)。在此虽然仅绘示一个实体储存装置阵列 400 连接至储存虚拟化控制器 200，但实际应用时可使用一个以上的实体储存装置阵列 400，而且主机 10 也可为一储存虚拟化控制器。

储存虚拟化控制器 200 接受由该主机 10 传来的输出入请求及相关数据(控制讯号及数据讯号)，并执行此输出入请求，或是将此输出入请求映像至实体储存装置阵列 400，而实体储存装置阵列 400 包含有多个实体储存装置（PSD）420，这些实体储存装置 420 可为例如硬盘。储存虚拟化控制器 200 可用来加强效能和/或改进数据安全性(data availability)，或是用来增加对主机 10 而言的单一逻辑介质单元的储存容量。

当储存虚拟化子系统 20 的逻辑介质单元的磁盘阵列为 RAID 0 或 RAID 1 以外的型态(例如 RAID3 至 RAID5)时，实体储存装置 400 中会包含有至少一奇偶校验实体储存装置 420，也就是说，此一实体储存装置 420 会存放有奇偶校验数据(parity data)，故整体的数据安全性因而提升。而且由于所处理的数据会被分送至不只一个实体储存装置 420，所以执行输出入操作的效能亦会有所提升。另外由于逻辑介质单元为多个实体储存装置 420 的结合，所以一单一逻辑介质单元中的可读储存容量亦可大幅提升。举例来说，RAID 5 的磁盘阵列子系统即可实现上述所有的功能。

当储存虚拟化子系统 20 的一逻辑介质单元设定为使用 RAID 1 时，相同的数据会被储存在两个实体储存装置 420 中。如此一来，虽然使实体储存装

置 420 的成本增加了两倍，但却可大幅提升数据的安全性(availability)或存取效率。

另外，当效能提升的重要性大于数据的安全性时，储存虚拟化子系统 20 的一逻辑介质单元可以设定为 RAID 0，此时数据安全性并不会因而提升，然而效能却可以有大幅的提升。例如一采用 RAID0、且有两个硬盘的磁盘阵列子系统，其相较于一般仅有一个硬盘的储存装置，所能提升的效能其理论值可达 200%，因为不同的数据段可经由储存虚拟化控制器 200 的控制，而同时储存入两个分开的硬盘。

请参考图 4，图 4 为连接至主机 10 及实体储存装置阵列 400 的储存虚拟化控制器 200 的一实施例方块图。此实施例中，储存虚拟化控制器 200 包含有一主机端输出入装置连结控制器 220，一中央处理电路 240 (central processing circuit)，一存储器 280，及一 SATA 输出入装置连结控制器 300。此处虽以分开的功能方块描述，但在实际应用时，两个以上，甚至全部的功能方块(functional block)可皆整合在一单一芯片上。

主机端输出入装置连结控制器 220 连接至主机 10 及中央处理电路 240，用来作为储存虚拟化控制器 200 及主机 10 之间的接口及缓冲，其可接收由主机 10 传来的输出入请求和相关数据，并且将其转换和/或对映至中央处理电路 240。主机端输出入装置连结控制器 220 可以包含有一个或多个用来耦接于该主机 10 的主机端端口。此处所提及的端口的类型可以为：光纤信道支持 Fabric (Fibre Channel supporting Fabric)、点对点连结、公用回路连结和/或专用回路连结于目标模式，操作于目标模式的平行 SCSI，支持因特网 SCSI(Internet SCSI; iSCSI)协议且操作于目标模式的以太网络，操作于目标模式的序列附加 (serial-attached) SCSI(SAS)，以及操作于目标模式的 SATA。

当中央处理电路 240 接收到来自主机端输出入装置连结控制器 220 的主机输出入请求时，中央处理电路 240 会将此输出入请求剖析，并且执行一些操作以响应此输出入请求，以及将所请求的数据和/或报告和/或信息，由储存虚拟化控制器 200 经由主机端输出入装置连结控制器 220 传送至主机 10。

将主机 10 传入的输出入请求剖析之后，若所收到的为一读取请求且一个或多个操作被执行以为响应时，中央处理电路 240 会由内部或由存储器 280 中或藉由此二种方式取得所请求的数据，并将这些数据传送至主机 10。

若所请求的数据无法在内部取得或并不存在于存储器 280，该读取请求将会经由 SATA 输出入装置连结控制器 300 发送至实体储存装置阵列 400，然后这些所请求的数据将由实体储存装置阵列 400 传送至存储器 280，之后再经由主机端输出入装置连结控制器 220 传送到主机 10。

当由主机 10 传入的写入请求(write request)传达至中央处理电路 240 时，在写入请求被剖析并执行一个或多个操作后，中央处理电路 240 通过主机端输出入装置连结控制器 220 接收从主机 10 传入的数据，将其储存在存储器 280 中。对于同步或异步装置操作两者，数据皆经由中央处理电路 240 传送至实体储存装置阵列 400。当该写入请求为一回写请求(write back request)，写入做完报告(IO complete report)会先被传送至主机 10，而后中央处理电路 240 才会执行实际的写入操作；而当该写入请求为一完全写入请求(write through request)，则写入做完报告会在数据已实际写入实体储存装置阵列 400 后才被传送至主机 10。

存储器 280 连接于中央处理电路 240，其作为一缓冲器，用来缓冲传送在主机 10 及实体储存装置阵列 400 之间通过中央处理电路 240 的数据。实际应用时，存储器 280 可以是 DRAM(动态随机存取存储器 Dynamic Random Access Memory)，该 DRAM 亦可为 SDRAM(同步动态随机存取存储器 Synchronous Dynamic Random Access Memory)。

SATA 输出入装置连结控制器 300 为介于中央处理电路 240 及实体储存装置阵列 400 间的装置端输出入装置连结控制器，用来作为一储存虚拟化控制器 200 及实体储存装置阵列 400 间的接口及缓冲。SATA 输出入装置连结控制器 300 接收由中央处理电路 240 传入的输出入请求及相关数据，并将其映像和/或传送至实体储存装置阵列 400。为了符合 SATA 协议的规范，SATA 输出入装置连结控制器 300 会将经由中央处理电路 240 传入的数据及控制讯号再格式化，并且将这些数据及讯号传送至实体储存装置阵列 400。

在本实施例中，附加于中央处理电路 240 的箱体管理服务电路 360(enclosure management service circuitry)是用来管理及监控储存虚拟化子系统 20 中的装置，这些装置包含但不限于有电源供应器、风扇、温度感知器、电压、不断电系统、电池、发光二极管(LED)、声响警报器、实体储存装置匣锁(PSD canister locks)以及门锁(door lock)。然而储存虚拟化子系统 20 亦有其它的配置方式，例如可依各种不同产品的功能设计而

定，而将箱体管理服务箱体管理服务电路 360 省略，或是将箱体管理服务电路 360 整合在中央处理电路 240 中。有关箱体管理服务(EMS)将阐述于其后。

请参考图 5，图 5 为中央处理电路 240 的一实施例，其中包含有 CPU 芯片组/奇偶校验引擎 224(CPU chipset/parity engine)，一中央处理器 242(CPU)，一只读存储器 246(ROM, read only memory)，一非易失性随机存取存储器 248 (NVRAM, non-volatile random access memory)，一液晶显示(LCD)模块 350，及一箱体管理服务电路 360。其中该 CPU 242 可为，例如，一 Power PC CPU，而 ROM 246 可为一闪存，用来储存基本输出入系统(BIOS)和/或其它程序。NVRAM 248 用来储存该实体储存装置阵列输出入操作执行状态的相关数据，以备输出入操作尚未做完前发生不正常电源关闭时，作检验使用。LCD 模块 350 则是用来显示子系统的操作状态，箱体管理服务电路 360 用来控制该直接存取控制器阵列的电源及进行其它的管理。ROM 246, NVRAM 248, LCD 模块 350 及箱体管理服务电路 360 皆经由一 X-总线(X-bus)连结至 CPU 芯片组/奇偶校验引擎 224。另外，该 NVRAM 248 及 该 LCD 模块 350 为可选择项目，在本发明的另一种配置中可以省略不设。

图 6 为本发明中 CPU 芯片组/奇偶校验引擎 224 的一实施例，CPU 芯片组/奇偶校验引擎 224 包含有奇偶校验引擎 260，CPU 接口 910，存储器接口 920，周边组件连结(Peripheral Component Interconnect; PCI)接口 930、932，X-Bus 接口 940，及主要存储器(Primary Memory; PM)总线 950，其中 PM 总线 950，举例而言，为一 64-bit，133Mhz 总线，且连接至奇偶校验引擎 260、CPU 接口 910、存储器接口 920、PCI 接口 930、932、X-Bus 接口 940 上，用以于其间通联数据讯号及控制讯号。

由主机端输出入装置连结控制器 220 所发出的数据及控制信号经由 PCI 接口 930，传送至 PM 先进先出缓冲器 934(PM FIFO)中缓冲，再进入 CPU 芯片组/奇偶校验引擎 224。其中连结至主机端输出入装置连结控制器 220 的 PCI 接口 930 可为，举例而言，64-bit，66Mhz。于 PCI 从属周期(PCI slave cycle)中，PCI 接口 930 拥有 PM 总线 950(PM Bus)，使 PM 先进先出缓冲器 934 中的数据及控制信号被传送至存储器接口 920 或是 CPU 接口 910。

由 PM Bus 950 传至 CPU 接口 910 的数据及控制信号，而后会传送至 CPU 242 进行处理，而 CPU 接口 910 及 CPU 242 间的沟通管道举例而言，可为 64-bit 数据传输线及 32-bit 地址线来进行。此数据及控制信号会经由一频宽为

64-bit, 133Mhz的CPU至存储器先进先出缓冲器922(CM FIFO; CPU to Memroy FIFO)，传送至存储器接口920。

在CM先进先出缓冲器922及存储器接口920之间，有一除错码产生电路924(ECC circuit, error correction code circuit)，用以产生一ECC码，而其产生的方式可为，举例而言，将8-bit的数据以异或(XOR)运算后，产生一单一位的ECC码。接下来，存储器接口920将数据及ECC码储存在存储器280中。该存储器280可为，举例而言，SDRAM。而存储器280中的数据经过除错码修正电路926(ECC correction circuit)，并与除错码产生电路924中的ECC码作比较，最后再被传送到PM Bus 950，其中除错码修正电路926是用来进行单一位自动修正(1-bit auto-correction)及多位检错(multi-bit error detecting)。

奇偶校验引擎260响应于CPU 242的指示，执行一特定磁盘阵列型态的奇偶校验功能。当然，在一些特定的条件下，比如说RAID0，奇偶校验引擎260会停止作动并不执行奇偶校验功能。在图6所示的实施例中，奇偶校验引擎260包含有一经由XOR先进先出缓冲器(XOR FIFO)264而连接至PM Bus 950的XOR引擎262,XOR引擎262将对一给定的地址及长度的存储器位置来执行XOR运算。

锁相回路980(PLL, phase locked loop)是用于在相关讯号间维持适当的相移(phase shift)。而计时控制器982 (timer controller)是用来提供各种不同讯号的时间基准。内部缓存器984 (internal register)是用来暂存CPU芯片/奇偶校验引擎224的状态，及控制PM Bus 950中的数据流动，而一对通用异步收发器(Universal Asynchronous Receiver and Transmitter, UART)功能方块986 (UART functionality block)则是用作CPU芯片/奇偶校验引擎224对外的接口，且该接口规格为RS232。

在实际应用时，PCI接口930，932可代换为周边组件连结扩充(Peripheral Component Interconnect eXtended; PCI-X)接口，或者是以周边组件连结快捷(PCI Express)接口取代PCI接口930，932。

请参考图7，图7为图4中SATA输出入装置连结控制器300的方块图，在本实施例中，SATA输出入装置连结控制器300包含有两个PCI-X至SATA控制器310 (PCI-X to SATA controller)。图8为图7中PCI-X至SATA控制器310的方块图，其中每个PCI-X至SATA控制器310包含有一连接至中

央处理电路240的PCI-X接口312，一连接至PCI-X接口312的译码／多任务(Dec/Mux)仲裁器314 (Dex/Mux arbiter)，以及八个连接至Dec/Mux仲裁器314的SATA端口600。PCI-X接口312包含有一连接至Dec/Mux仲裁器314的总线接口318，以及一用来储存PCI-X至SATA控制器310组态的组态电路316 (configuration circuit)。Dec/Mux仲裁器314将在PCI-X接口312与多个SATA端口600间进行仲裁，且执行自PCI-X接口312至SATA端口600的交易(transaction)的地址译码。而数据及控制讯号将经由此PCI-X至SATA控制器310的SATA端口600，被传送至实体储存装置420。在实际应用中，PCI-X至SATA控制器310可由PCI至SATA控制器取代，而在PCI至SATA控制器中，PCI-X接口312可由一PCI接口取代。同样地，在其它的实施例中，PCI-X至SATA控制器310可由一PCI Express至SATA控制器取代，而在PCI Express至SATA控制器中，PCI-X接口312是由一PCI Express接口取代。

接下来请参考图9，图9为图8中SATA端口600的一实施例方块图。如图9中所示，SATA端口600包含有一超集缓存器630 (superset register)，一指令区块缓存器640 (command block register)，一控制区块缓存器650 (control block register)，一直接存储器存取缓存器620 (DMA register)。经由上述的缓存器以及通过一由直接存储器存取控制器670所控制的双端口先进先出缓冲器660，数据得以在Dec/Mux仲裁器314与传输层690 (transport layer)间传输。数据传送至传输层690后，会被再格式化成为帧信息结构 (FIS, frame information structure)，并传送到连结层700 (link layer)。

连结层700稍后将帧信息结构转化成为帧(frame)，以加入帧起始信息(Start Of Frame, SOF)，循环冗余校验码(Cyclic-Redundancy Check Code, CRC)，帧结束信息(End Of Frame, EOF)等，并将其以8b/10b编码方式转译成8b/10b编码的字符而实现，并将其传送到实体层710 (PHY layer)。

实体层710经由一对差动讯号线(differential signal lines)—传输线LTX+及LTX- —传送出讯号至实体储存装置420，并经由另一对差动讯号线—接收线LRX+及LRX- —接收来自实体储存装置420的讯号，其中各组的两条讯号线，例如LTX+及LTX-，同时个别传送以一参考电压Vref为准的正负电压的讯号TX+/TX-，例如+V/-V或是-V/+V的电压讯号，所以它们的电压

差是+2V或是-2V，如此一来便可增加讯号的品质。在LRX+及LRX-接收线上也可以使用相同的方法接收讯号RX+/RX-。

当一帧由实体层710传送至连结层700，连结层700将用8b/10b编码的字符进行译码，并且除去SOF，CRC，EOF的部份，其中经由帧信息结构FIS计算得出的CRC将会被拿来与所接收到CRC作比较，用来确定所接收的信息的正确性。当传输层690接收到来自连结层700的FIS讯号，传输层690将会决定FIS的型式，并依照FIS的型式将FIS的内容传送到所指定的区域。

图10为符合SATA协议的传输结构，其中在序列线中通联的讯号为一连串使用8b/10b编码的字符，其最小单位为双字组(double-word，32位)。每一个双字组的内容将被组合以提供低阶的控制信息，或是用以传送主机与相连结的装置间的信息，而在讯号线上传送的两种信息结构为基元(primitive)以及帧。

一基元是由一单一的双字组所组成，其为主机与装置间通讯信息中最简单的单位。当一基元中的字节在编码之后，其所产生的型样(pattern)便不太会被误解成其它型式的基元或是其它任意的型态。基元主要的用途是传送实时(real-time)状态的信息，这些信息是用来控制信息的传递以及协调主机及装置间的通讯。一基元的第一字节为一特别的字符。

一帧是由多个双字组所构成，并以SOF(Start of Frame)基元开始，以EOF(End of Frame)基元结束。在SOF基元之后为一使用者有效负载，称之为FIS (帧信息结构Frame Information Structure)。另外CRC(循环冗余校验码Cyclic-Redundancy Check Code)为紧接在EOF基元之前的最后非基元双字组，且CRC为依据FIS运算得来。另外，介于SOF与EOF间的流程控制基元HOLD或是HOLDA是用来调整数据流，以实现速率匹配 (speed matching)的目的。

传输层690用来在传送时建构FIS，或者是在自连结层700接收到FIS时将其分解。且该传输层690并不维护ATA指令或是先前的FIS内容的前后关系(context)。当收到请求时，传输层690会收集FIS内容，并依照正确的顺序建构FIS。FIS的型态有很多种，图11及12分别为其中之一。

如图11所示，一直接存储器存取设定的FIS在字段0处包含有一标头(HEADER)，而其第一字节(字节0)则定义了该FIS的型态(41h)，此FIS的型态则定义了此FIS其余的字段，和定义它的全部长度为七个双字组。字节1

中的位 D 则标示了该后续数据传送的方向，D 为 1 表示传送端至接收端，D 为 0 则表示接收端至传送端。字节 1 中的位 I 为一中断位(interrupt bit)，而位 R 为一保留位，且设为 0。直接存储器存取缓冲器识别码的高/低字段(DMA buffer identifier high/low field，字段 2 和字段 1)，则分别标示了该主机存储器的直接存储器存取缓冲区域。直接存储器存取缓冲器偏移栏(DMA buffer offset field，field 4)，为进入缓冲器内的字节偏移。直接存储器存取传送计数栏(DMA transfer count field，field 5)，则为此装置所读取或写入的字节数量。

如图 12 所示，一数据型 FIS(DATA FIS)在字段 0 处包含有一标头，且该第一字节(字节 0)定义了该数据型 FIS 的型态(46h)，而数据型 FIS 的型态则定义了其余的字段以及它的全长为 n+1 双字组(double-word)。还有，字节 1 中的多个 R 位则为保留位，并且设定为 0，而字段 1 至 n 中的数据为双字组，并包含有将被传送出的数据。数据型 FIS 的数量有其最大上限的限制。

亦即，图 4 中所示的装置端输出入装置连结控制器(SATA 输出入装置连结控制器 300)会将所接收到的装置端输出入请求与伴随的输出入数据再格式化成一传输用的数据包，这些数据包包含有一用来指示该数据包起始端且位于前端的起始段 (start segment) 、一用来指示该数据包终结端且位于尾端的终结段 (end segment) 、一经由该装置端输出入装置连结传送且含有实际输出入信息的有效负载数据段 (payload data segment) 、以及一含有由该有效负载数据导出并用来检核传送后的有效负载数据正确性的检核码(check code) 的检核数据段 (check data segment) 。

在图 4 的实施例中，主机端输出入装置连结控制器 220 及装置端输出入装置连结控制器 300(SATA 输出入装置连结控制器 300)，可使用相同类型的 IC 芯片，而其中主机端输出入装置连结控制器 220 上的输出入装置连结端口的组态被设定为主机端的输出入装置连结端口，而装置端输出入装置连结控制器 300 中的输出入装置连结端口的组态则被设定为装置端的输出入装置连结端口使用。另外，亦可采用一单一芯片，其组态可被设定为同时包含有主机端输出入装置连结端口及装置端输出入装置连结端口，用以在同一时间分别耦接至主机 10 及实体储存装置阵列 400。

接下来将介绍主机 10 及储存虚拟化控制器 200 间的输出入流动的实例，以及储存虚拟化控制器 200 与实体储存装置 420 间的输出入流动的实例。请

参阅图 13 及 14，图 13 及 14 为主机 10 及储存虚拟化控制器 200 间的输出入流动的实例。输出入请求是从主机 10 经由主机端输出入连结传入，且此输出入请求将被剖析解读以决定所要执行的操作，同时对于同步装置操作以及异步装置操作，决定这些操作在哪些逻辑介质单元的区段上执行。若这些操作仅包含有内部仿真操作及异步装置子操作，则储存虚拟化控制器 200 执行这些相关的子操作，这些子操作包含有传送任何相关数据至主机 10（或自主机 10 接收相关数据），并且连带传送有一状态报告，以告知主机 10 该操作是成功或是失败，以及失败的相关原因。若这些操作包含有同步装置操作，则会产生适当的装置端输出入请求并送至适当的实体储存装置 420 以为响应，而各装置端输出入请求的内容及其传送的目标实体储存装置 420 则依照与特定的逻辑介质单元相关的特定的映像方式所决定。在装置端输出入请求执行的同时或之前，任何将从主机 10 中获得以当作主机端输出入请求执行的一部份且其后来会被传送到实体储存装置 420 的有效负载(payload)数据，将会由主机 10 传送到储存虚拟化控制器 200 中。

在装置端输出入请求成功做完之时，响应于该装置端输出入请求所读取的数据将会被传送到送出请求的装置，若在快取架构中该装置可为该快取，且由该主机所请求的任何数据将传送至主机 10。其后并产生一状态报告传送至主机 10 以告知操作已成功做完。如果有装置端输出入请求无法被成功做完，储存虚拟化控制器 200 将会启用备用操作，这些备用操作其即便在面临个别的装置端输出入请求失败的情况下仍会成功做完其未成功做完的子操作。这些操作典型地包含产生其它的装置端输出入请求至不同的介质区段(media section)，以在读取的情况下回复所欲读取的数据，或是在写入的情况下写入备份数据(backup data)。RAID 5 就是其中一例，若一特定的实体储存装置 420 读取失败时，其可利用存于其它实体储存装置 420 中的数据，重新产生该所欲读取的数据。另一方面，储存虚拟化控制器 200 也可以选择不完成该子操作，停止数据传送到主机 10，且回传一相应的状态报告至主机 10。

请参考图 15 及 16，图 15 及 16 为储存虚拟化控制器 200 与实体储存装置 420 间的输出入流动的流程图。对于每个产生来作为同步装置子操作的装置端输出入请求，其输出入请求信息定义有该特定输出入操作的各类参数，如：目的介质区段基准地址(destination media section base address)、

介质区段长度(media section length)、指示所要执行的操作的命令(command indicating operation to be performed)等等，这些输出入请求的信息将会再格式化成缓存器主机至装置型态(Register-Host-to-Device)的帧信息结构FIS，并封包成一SATA帧，然后再经由SATA连结传送到相关的实体储存装置420，其中每一帧包含有一由帧中的数据所计算得来的CRC值，以致于假如传送到实体储存装置420的帧中的任一数据在传送途中被变更，则实体储存装置420可藉由于接收到帧之后执行一致性检核而获知，该一致性检核为计算所接收的帧中数据的CRC值并与帧内所包含的该CRC值比对。当该CRC值的比对不相符时，该实体储存装置420便会在接收该帧之后传送一R_ERR基元至储存虚拟化控制器200，显示其所接收到的帧数据发生更动。而后储存虚拟化控制器200可依其选择，而再传送一次该帧，或是中止该交易(transaction)并回复一相关的状态报告至发出请求的实体。

假如接收到的帧是完整而未经更动的，实体储存装置420便会在接收该帧之后回传一R_OK基元至储存虚拟化控制器200，以通知储存虚拟化控制器200该帧被完整接收而未经更动。实体储存装置420将包含在帧中的请求进行剖析，并决定所要执行的操作性质及所在的介质区段。假如在所决定出的操作在该特定介质区段上不是一个有效操作、或是所指定的介质区段为无效的区段，则实体储存装置420将会响应给该储存虚拟化控制器200以一相应的状态报告；此是藉由产生一包含有状态信息的SATA缓存器装置至主机型态的FIS，将此FIS封包成一SATA帧并传送回储存虚拟化控制器200而实现。否则，该实体储存装置将会执行该操作。

在执行操作的同时或之前，假如还必须自储存虚拟化控制器200传送有效负载数据至实体储存装置420，则实体储存装置420将产生并发出一SATA帧，该SATA帧将传送一直接存储器存取启动装置至主机型态FIS(DMA-Activate-Device-to-Host FIS)，请求传送第一组数据。储存虚拟化控制器200将会把数据分解成许多个区块，而这些区块最大的长度不得超过SATA协议当中单一帧的最大长度，其中每一区块将会封包成数据主机至装置型态FIS(Data-Host-to-Device FIS)的帧，并且逐个的传送到实体储存装置420。在每一个帧传送之后，储存虚拟化控制器200会等待接收来自实体储存装置420的一个用以递送一直接存储器存取启动装置至主机型态FIS的帧，表示实体储存装置420在传送下一个帧的数据之前已经准备好要接收更

多的数据。每个帧的数据包含有一由该数据中产生的 CRC 值，实体储存装置 420 将会对每一个帧检查该 CRC 值与帧中的数据的一致性，若发生不一致时，该输出入操作就会被中止，而实体储存装置 420 会产生一相关的状态报告，该状态报告的产生是藉由产生一包含有状态信息的 SATA 缓存器装置至主机型态 FIS，并将其封包成一 SATA 帧，并传回至储存虚拟化控制器 200 而完成。储存虚拟化控制器 200 收到状态报告时，可以依其选择，重新再发送该原来的输出入请求来再试一次该操作，或者可以中止该交易，并回传一状态报告至发出请求的实体。

在操作执行期间，以及/或是操作执行做完后，假如需要将有效负载数据自实体储存装置 420 传送到储存虚拟化控制器 200，则实体储存装置 420 会将数据准备好(可能需要从储存介质中将数据读出)，并且会把数据分解成许多个区块，而这些区块最大的长度不得超过 SATA 协议当中单一帧的最大长度，其中每一区块将会封包成数据装置至主机型态 FIS(Data-Device-to-Host FIS)的帧，并且逐帧地传送到储存虚拟化控制器 200。再一次地，每个帧包含有一由此帧的数据产生的 CRC 值，且该 CRC 值会在该帧中而被传送至储存虚拟化控制器 200，该储存虚拟化控制器 200 在收到每一个帧时，将会检查该 CRC 值与帧内数据的一致性，若该接收帧内的数据所计算出的 CRC 值跟帧中所传送的 CRC 值不一致时，储存虚拟化控制器 200 在收到帧后会传送一 R_ERR 基元至实体储存装置 420 以为响应，告知所接收的帧发生变更。而实体储存装置 420 典型地将会立即中止该输出入操作，并产生一相关的状态报告，该状态报告的产生是藉由产生一包含有状态信息的 SATA 缓存器装置至主机型态 FIS，并将其封包成一 SATA 帧，并传回至储存虚拟化控制器 200 而实现。储存虚拟化 200 收到状态报告时，可以依其选择，重新再发送该原来的输出入请求来再试一次该操作，或者可以中止该交易，并回传一状态报告至发出请求的实体。

假如接收到的帧是完整的而未经更动，储存虚拟化控制器 200 便会对每一个数据装置至主机型态 FIS 帧响应以一 R_OK 基元。当所有身为该输出入请求执行的一部份的要递送的数据皆被传送到储存虚拟化控制器 200 时，实体储存装置 420 将会产生一状态报告，指出该操作是成功做完或是失败，以及失败的原因，其中该状态报告被格式化为一 SATA 缓存器装置至主机型态 FIS (Register-Device-to-Host FIS)，并被封包成一 SATA 帧，且发送回储

存虚拟化控制器 200。接下来，储存虚拟化控制器 200 将此状态报告进行剖析以决定输出入请求为成功或失败，假如状态报告其为输出入请求失败，储存虚拟化控制器 200，可以依其选择，重新再发送该原来的输出入请求，来再试一次该操作，或是可以中止该交易，并回传一相应的状态报告至发出请求的实体。

至于传统的 PATA 储存虚拟化控制器的流程，则和上述 SATA 储存虚拟化控制器类似，不同之处在于，该定义输出入操作参数(如目的介质区段基准地址，或介质区段长度等)的原来的装置端输出入请求信息，并没有被封包成一个所运送的数据会经过有效性检核的帧，而与本发明的 SATA 利用帧的 CRC 值来确认其有效性不同。所以当数据在由储存虚拟化控制器至实体储存装置间的传输上，不慎发生损坏，比如说受到噪声的影响，则将无法被检测出来。此将可能造成灾难性的数据破坏情况，因为如果原来的输出入请求数据中的目的介质区段基准地址以及/或是介质区段长度若因为毁损而有误的话，将导致数据可能写入到错误的介质区段。在 SATA 的应用当中，上述可能发生的破坏、毁损错误都可由帧的 CRC 值来检测出，因为该帧的 CRC 值会与数据发生不一致，而该实体储存装置将会中止该命令而不将数据写入至错误的介质区段或自错误的介质区段读出。这是在储存虚拟化控制器上施行 SATA 架构相较于 PATA 架构的最主要的好处。

实际应用时，假如有多个主机端输出入请求，储存虚拟化控制器 200 会在同时进行多个操作，这些输出入请求有可能是从一个，抑或是多个主机所发出，其中这些操作可能包含有同时执行的同步装置子操作，而这些子操作每一个可以产生多个装置端输出入装置请求寻址到不同的实体储存装置 420。此每一个输出入装置请求可能需要在储存虚拟化控制器 200 与所寻址的实体储存装置 420 间通过连接于两者间的装置端输出入装置连结来传送大量的数据。通常储存虚拟化控制器 200 都会被设定组态成可以将此种输出入装置请求分散至不同的实体储存装置 420 及不同的装置端输出入装置连结上，以使得实体储存装置 420 与输出入装置连结的集合频宽(collective bandwidth)最大化。其中一个组态设定上可改善频宽最好的例子，为使用 RAID 5 方式而不是采用 RAID4 的方式将实体储存装置 420 结合为逻辑介质单元。在 RAID 4 的架构中具有一专用的奇偶校验硬盘用以储存所有的奇偶校验数据。在写入数据时，每一笔数据在写入动作时都会要求写入一更新的奇

偶校验数据，如此一来，奇偶校验磁盘将远较其它数据硬盘还要忙碌。而在读取数据时，此奇偶校验硬盘将没有被存取，亦即会有一个硬盘对于此传送数据的任务中没有贡献。而 RAID5，它的奇偶校验数据是分布在所有的磁盘中，所以，在假设输出入请求是平均的寻址于各个实体储存装置，那么在写入动作时将不会有其中一个硬盘比其它硬盘忙碌的情况发生，同时，在读取动作时于传送数据的任务中所有硬盘都有其贡献。

此外，储存虚拟化控制器 200 还可以使用智能型的机制，来动态调整不同实体储存装置 420 和/或输出入装置连结间的输出入装置请求的分配，以期将实体储存装置/连结集合频宽进一步地最佳化。其中一个例子是连接到同一组实体储存装置 420 的输出入连结间的负载平衡，该储存虚拟化控制器 200 会智能型地一直追踪经由各个连结所传送的输出入装置请求，而从这些追踪的数据来决定下笔输出入装置请求应由哪个连结来传送，以将连结的集合频宽最大化。另一个例子则是一组镜射的（mirrored）实体储存装置 420 间的数据读取输出入请求的负载平衡，再次地，储存虚拟化控制器 200 会智能型地纪录寻址至每一实体储存装置 420 的输出入装置请求，来决定下笔输出入装置请求应送到哪里，以使镜射的实体储存装置 420 组的集合频宽最大化。

将集合频宽最大化之后，一个以储存虚拟化控制器扮演主要角色的储存虚拟化子系统的总体效能，在一些型式的主机输出入请求负载情况下，将受到最大集合频宽的限制。在这种状况下，增加集合频宽可以提升其效能。一般来说，总体的装置端输出入装置连结效能是由两个因素决定的：此连结中的输出入请求执行/数据传输率，以及输出入请求/数据传输所能通过的连结数目。连结的输出入请求执行/数据传输率越好，总体效能自然越高，同样地，输出入请求/数据传输所能通过的连结越多，装置端输出入连结子系统的整体效能越好。

如前所述，PATA 受限于其形成单一个别连结所需的专用讯号线数目很多(28)，因而当 PATA 连结的数目在超过某一点后就不易增加。因此，典型的 PATA 储存虚拟化控制器可以包含不超过 12 个装置端 PATA 输出入装置连结。平行 SCSI 不仅有每一个连结有 68 个讯号线的缺点，它相较于 PATA 或是 SATA 而言，每一个连结的昂贵价格以及在印刷电路板(printed circuit board)上所占用的庞大面积更为其一大缺憾。一典型的储存虚拟化可使用 4 至 8 个

独立的装置端平行 SCSI 输出入装置连结，其中单一个连结的价格可能就已是一个 PATA 或是一个 SATA 连结的好几倍。光纤连结由于印刷电路板上的面积很大，以及每个连结的单位成本很高(通常要较 PATA/SATA 高出一个数量级)，也是使其连结的数目不易增加的原因。

SATA 输出入连结的数目则较易增多，因为每一个连结仅由四条讯号线所组成，且可以进行高度的整合，使一单一 SATA 控制 IC 可支持 8 个连结（以数量相当的插脚数目及大小而言，标准平行 SCSI 以及光纤都只能支持两个连结）。而且，SATA 的每个连结都具有相对较低的成本，所以一单一有效降低成本的储存虚拟化控制器可包含有许多的装置端 SATA 输出入装置连结。

所有的装置端输出入装置连结都是 SATA 的纯 SATA 储存虚拟化控制器（pure SATA SVC controller）有一个限制，就是它的可连结的实体储存装置的数目受限于可包装在一单一储存虚拟化控制器当中的装置端输出入装置连结的数目，而 SATA 的规格当中，讯号线的最大长度仅限于 1.5 公尺，以致于连接到一储存虚拟化控制器的实体储存装置一定要靠的够近，使讯号线的长度不超过 1.5 公尺。由于这些限制，SATA 储存虚拟化子系统只能提供最多 16 个 SATA 实体储存装置的连接。所以一纯 SATA 储存虚拟化子系统无法像光纤 FC-AL 储存虚拟化子系统一样，拥有经由同一组装置端输出入装置连结的外接扩充机箱连接至最多为 250 个实体储存装置的扩充性。

为了克服以上的限制，本发明选择性地可包含一个或多个扩充装置端多装置输出入装置连结(expansion device-side multiple-device IO device interconnect)，在此称为装置端扩充端口，如储存虚拟化控制器上的平行 SCSI 或是光纤 FC-AL，而这些连结可允许外接扩充机箱(chassis)。这些机箱可为直接连接到连结上，而不需要介于其中的转换电路的 JBOD 实体储存装置，也可是智能型 JBOD 仿真子系统。此 JBOD 仿真子系统为使用 SATA 或 PATA 实体储存装置组合及单一或者冗余储存虚拟化控制器，其中的储存虚拟化控制器是用来提供将连接 JBOD 子系统与主要储存虚拟化子系统(primary storage virtualization subsystem)的多装置端输出入装置连结协议，转换到连接 JBOD 储存虚拟化控制器与其所管理的实体储存装置的装置端输出入装置连结(SATA 与 PATA)协议。

请参阅图 17，图 17 为支持装置端扩充端口的一储存虚拟化子系统的实施例。图 17 中，每一扩充端口所连接到的储存单元皆为单一端口，然而，

若该储存单位为双端口，具有一对或多对设定为冗余组态扩充端口的储存虚拟化控制器，则可将其一冗余扩充端口对中的一端口连接到储存单元中双端口对中的一端口，而此冗余扩充端口对中的另一端口则连接到此储存单元中双端口对的另一端口。图 18 即披露了这样的组态，其中，若储存虚拟化控制器的冗余扩充端口对的其中的一端口发生故障，或是储存单元中双端口对中的一端口发生故障，或是连接储存虚拟化控制器冗余扩充端口对与储存装置的双端口对输出入连结其中之一断掉或是被阻断的话，储存虚拟化控制器仍然可以对储存单元经由另一替代路径进行存取，此替代路径是由从此储存虚拟化控制器的另一替代端口连接至储存单元的替代端口的连结所构成。

SATA 储存虚拟化子系统也可以使用 PATA 实体储存装置而不使用 SATA 实体储存装置。这样在 PATA 实体储存装置旁边，需要安插一个 SATA 至 PATA 转换电路，而该转换电路是将 SATA 讯号及协议，转换成 PATA 讯号及协议，并在相反方向时再转换回来。虽然背板讯号线路(backplane signal trace)在 SATA 至 PATA 转换电路与实体储存装置间具有的一小段 PATA 讯号线路(signal trace)可能会有易于出现传送其间的信息未受保护而毁损的问题，但是此背板讯号线路因为其长度与其讯号线的数目（如前所述，PATA 在每一个连结尚须使用 28 条讯号线）而容易受噪声及串音（cross talk）效应影响的情形却由于使用到了 SATA 的讯号传输标准而获得保护，这是因为 SATA 改善了的错误检测能力使得数据免于发生未受保护的毁损之故。除此以外，实质上，所有 SATA 储存虚拟化子系统相较于采用其它标准装置端输出入装置连结的优点，都在本发明的实施例中实现。

在短期来说，SATA 磁盘的供应仍然短缺，且它的单价亦不算太便宜，因此在这段过渡期使用 PATA 实体储存装置来替代 SATA 实体储存装置用于一 SATA 储存虚拟化子系统中有其重要性。在此过渡期间，此种子系统让 PATA 实体储存装置可以替代 SATA 实体储存装置，消除了 SATA 实体储存装置供应上及成本上的顾虑。在这样的子系统中通常可将转换电路放置于存放实体储存装置的可拆卸匣(removable canister)之中。因此，当后续有实体储存装置或相关电路需要进行维修服务时，可以很容易地从系统上拆卸下来。此外，藉由将转换电路设置于可拆卸匣当中，在 SATA 实体储存装置价格降低至较可接受的程度时，原先装设 PATA 的可拆卸匣即可很方便的整个从系统中移除，并将新的 SATA 实体储存装置装及相关电路设置到系统上。

请参阅图 19 及 20，图 19 为可拆卸 PATA 实体储存装置匣的方块图，图 20 则为一可拆卸 SATA 实体储存装置匣的方块图，图 19 及 20 中都有来自于储存虚拟化控制器的 SATA 输出入装置连结。这二个图中主要不同的地方在于可拆卸 PATA 实体储存装置槽中多了一个 SATA 至 PATA 转换电路，这在可拆卸 SATA 实体储存装置槽中是没有的。同时，该 PATA 实体储存装置匣及其中的相关电路的设计，使得该 PATA 实体储存装置可以热插拔以及冷插拔；亦即，当该储存虚拟化子系统或该储存虚拟化控制器在线上(on-line)时或不在线上(off-line)时，该 PATA 实体储存装置可插入其中或自其中移除。同样地，该 SATA 实体储存装置匣及其中的相关电路的设计，使得该 SATA 实体储存装置可以热插拔以及冷插拔。

此外，在一个储存虚拟化控制器中还有可能使用冗余主机端连结架构。在此种架构下，储存虚拟化控制器中包含有多个主机端连结端口时，以将逻辑介质单元以相同的形式通过二个或更多的主机端连结呈现至主机。此种设计的特色是，即使其中一条主机端连结或端口出现断掉、阻断或故障，主机能仍继续对此逻辑介质单元进行存取。

在上述的架构当中，在储存虚拟化控制器中的两个个别的主机端端口，连接至两个完全分开的主机端输出入装置连结以及主机端口(图未示)。而一个在主机端连结支持冗余的架构中，储存虚拟化控制器会将同一逻辑介质单元以相同的形式呈现至此二个主机端口。

储存虚拟化子系统通常包含有由储存虚拟化控制器来监管子系统中，如电源供应器，风扇，温度感知器等等装置的功能，如前所提及，这种管理功能称为箱体管理服务(EMS)。箱体管理服务通常在实做上使用的是一种包含有 CPU 并执行一软件程序以实现所需功能的一种智能型电路(intelligent circuitry)。一般来说，平行 SCSI 及光纤储存虚拟化子系统是分别使用 SCSI 协议中的 SAF-TE(SCSI 存取容错箱体; SCSI Accessed Fault Tolerant Enclosures)及 SES (SCSI 箱体服务; SCSI Enclosure Services)，作为储存虚拟化控制器与储存虚拟化子系统的箱体管理服务间的主要通讯机制，而这些协议则是靠储存虚拟化控制器及由一用来传输 SCSI 指令协议的输出入装置连结间的连结来完成，如平行 SCSI 或光纤连结。但是，典型的 SATA 储存虚拟化子系统当中于储存虚拟化控制器及近端储存虚拟化子系统(local SVS)间并没有这样的连结存在，且这种连结实作上虽可实施，但无疑的却会

增加不少成本，所以寻求使用一个成本低廉的连结而且通过此连结使用专用的协议将会是比较低成本的方法。请注意，前面所述及的扩充端口，其是用来连接远程装置，如 JBOD 子系统，并非本处所提的近端储存虚拟化子系统(local SVS)。

I2C(Inter-IC bus;集成电路间总线)是一种低成本的连结，它可以支持双向数据传输于一可接受的传输速率下，常使用于 PC 中，令 CPU 得以管理与监控主机板与其它装置的状态，所以现在在储存虚拟化控制器与近端的箱体管理服务中使用这种连结是很合适的，尤其是在储存虚拟化控制器与储存虚拟化子系统间先前没有连结的 SATA 储存虚拟化子系统中。然而，依据标准，这种连结并不支持 SCSI 指令协议的传送，所以储存虚拟化控制器与近端储存虚拟化中的箱体管理服务间主要通讯媒介所使用的传输协议就必须要使用其它的协议，而且最好是专用的协议。

若使用 I2C 作为此主要通讯媒介，则箱体管理服务可以有下列两种实施方式。第一种是使用智能型电路，也就是使用和 SAF-TE/SES 相似的智能型协议(intelligent protocol)。而第二种是将现成的没有特别功能的 I2C 锁存和/或状态监控的 IC 整合成一个储存虚拟化控制器能解读的管理/监控电路，且让所有的智慧留给储存虚拟化控制器。前者有一个好处，它可以让箱体管理服务提供更多进阶服务的功能、具有更高的价值，且可客制化，但在实作上通常较为复杂也比较昂贵。后者虽然于实作上较简单也较便宜，但通常没办法支持进阶的功能。

储存虚拟化子系统中的实体储存装置子系统通常被设计成可用来模拟典型的箱体管理服务，而可为一主机通过一输出入装置连结来直接管理与监控，其中该输出入装置连结同时也是该实体储存装置在子系统中所使用的主要存取连结。实作中，箱体管理服务电路是智能型电路，并且使用标准 SCSI 协议来监控箱体管理服务，如 SAF-TE 及 SES，这些协议都是可以传输于主要存取连结(primary access interconnect)上的。箱体管理服务控制器(EMS controller)会直接连接到一个或多个主要存取输出入装置连结，来实现与主机间的直接通联，这种组态称为直接连结(direct-connect)，或者箱体管理服务控制器会通过直接连接到主要存取连结的装置(如 PSDs)所支持的转传机制(pass-through mechanism)，从主机来传送请求及相关数据至箱体管理服务控制器，并且从箱体管理服务控制器回传相关数据至主机，而这样的

组态称为装置代传(device-forwarded)。使用直接连结组态的箱体管理服务提供了与实体储存装置的独立性，将不会因一个甚或多个实体储存装置失效或不在而受到影响或无法存取，但是它却有昂贵以及过于复杂的缺点。装置代传组态的箱体管理服务在实作上较简单，而且低成本，但是仍有其缺陷，如当实体储存装置坏了或是不在，主机可能就没有办法继续使用箱体管理服务。

为了加强储存虚拟化子系统与设计来和实质实体储存装置子系统接口的主机的兼容性，装设了箱体管理服务的该存虚拟化子系统可以做成支持上述一个或多个标准 SCSI 箱体管理服务协议以及上述的直接连结和装置代传二组态中的一者或两者。在直接连结模拟中，该储存虚拟化控制器在一主机端输出入装置连结中呈现一个或多个识别码/逻辑单元号(logical unit number)，且箱体管理服务可以具有指定给它的专用连结识别码，或是它可以仅有被指定在识别码上的逻辑单元号且该识别码已经呈现有其它逻辑单元号。而 SAF-TE 模拟中，储存虚拟化控制器必须呈现箱体管理服务 SAF-TE 装置于专用的识别码上。在直接连接 SES 模拟中，箱体管理服务 SES 装置可以呈现于专用的识别码上或是在已呈现有其它逻辑单元号的识别码上呈现。而装置代传仿真中，储存虚拟化控制器仅会在负责代传箱体管理服务管理请求的虚拟实体储存装置的询问字符串(INQUIRY string)中包含一些数据，其中箱体管理服务管理请求会告诉主机该实体储存装置的功能之一是代传请求至该箱体管理服务。通常，多个或是全部呈现于连结上的虚拟实体储存装置将会成为箱体管理服务请求的代传者，因此一个或多个实体储存装置的不在或毁坏，不会造成无法存取箱体管理服务。

以上所述仅为本发明的较佳实施例，凡依本发明的权利要求所做的均等变化与修饰，皆应属本发明专利的涵盖范围。

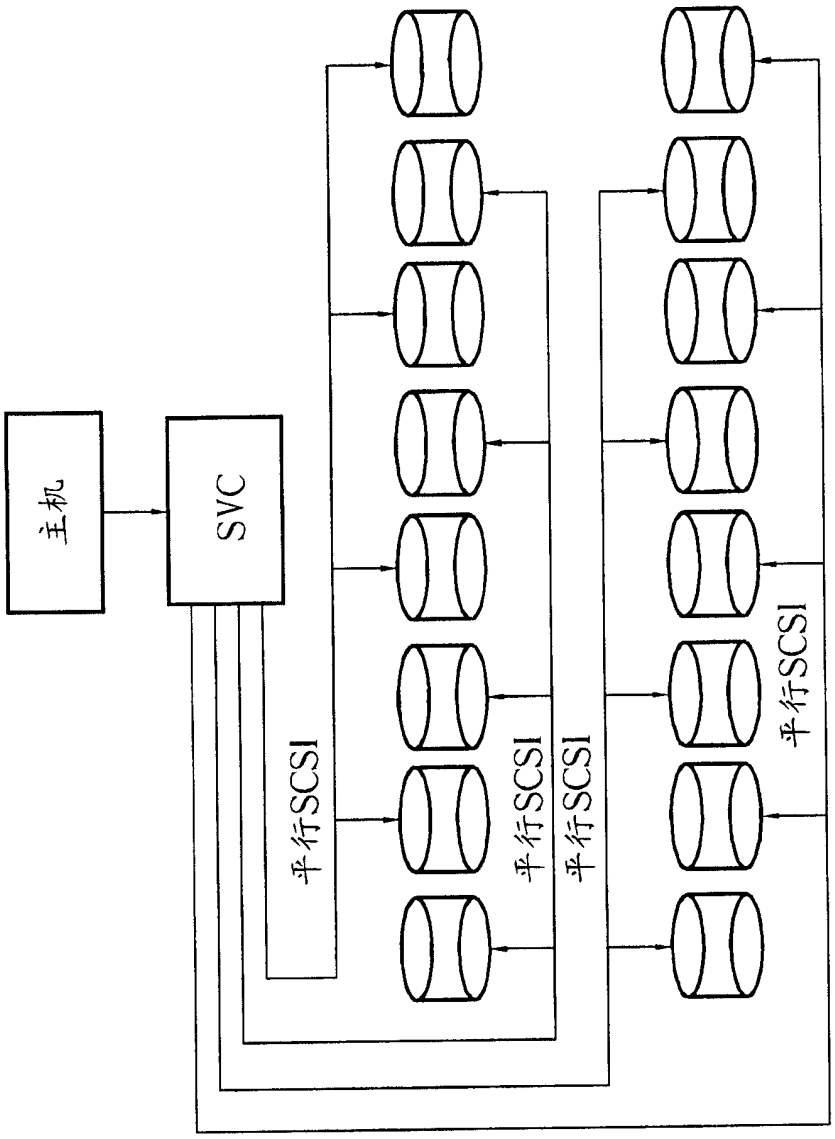


图 1

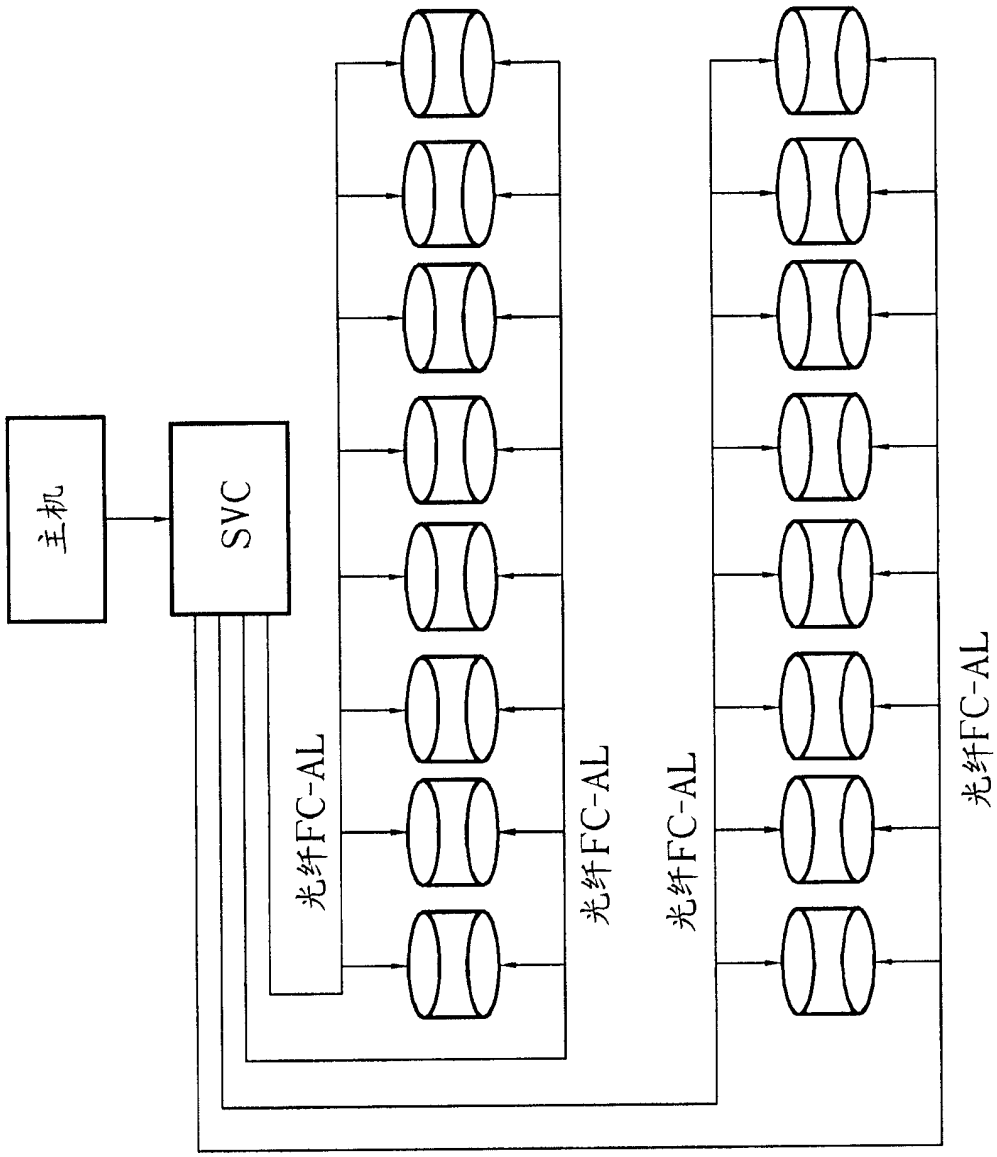


图 2

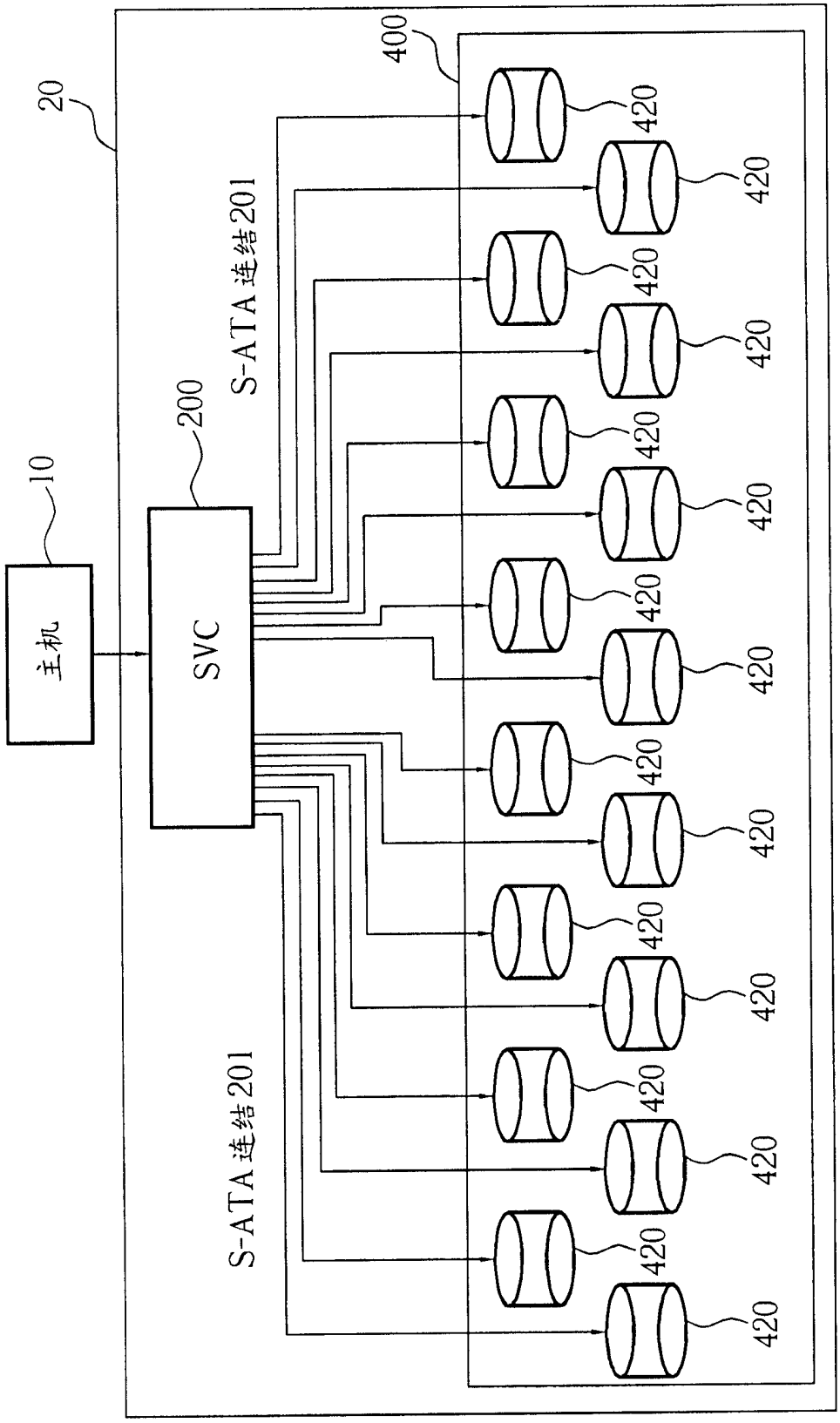


图 3

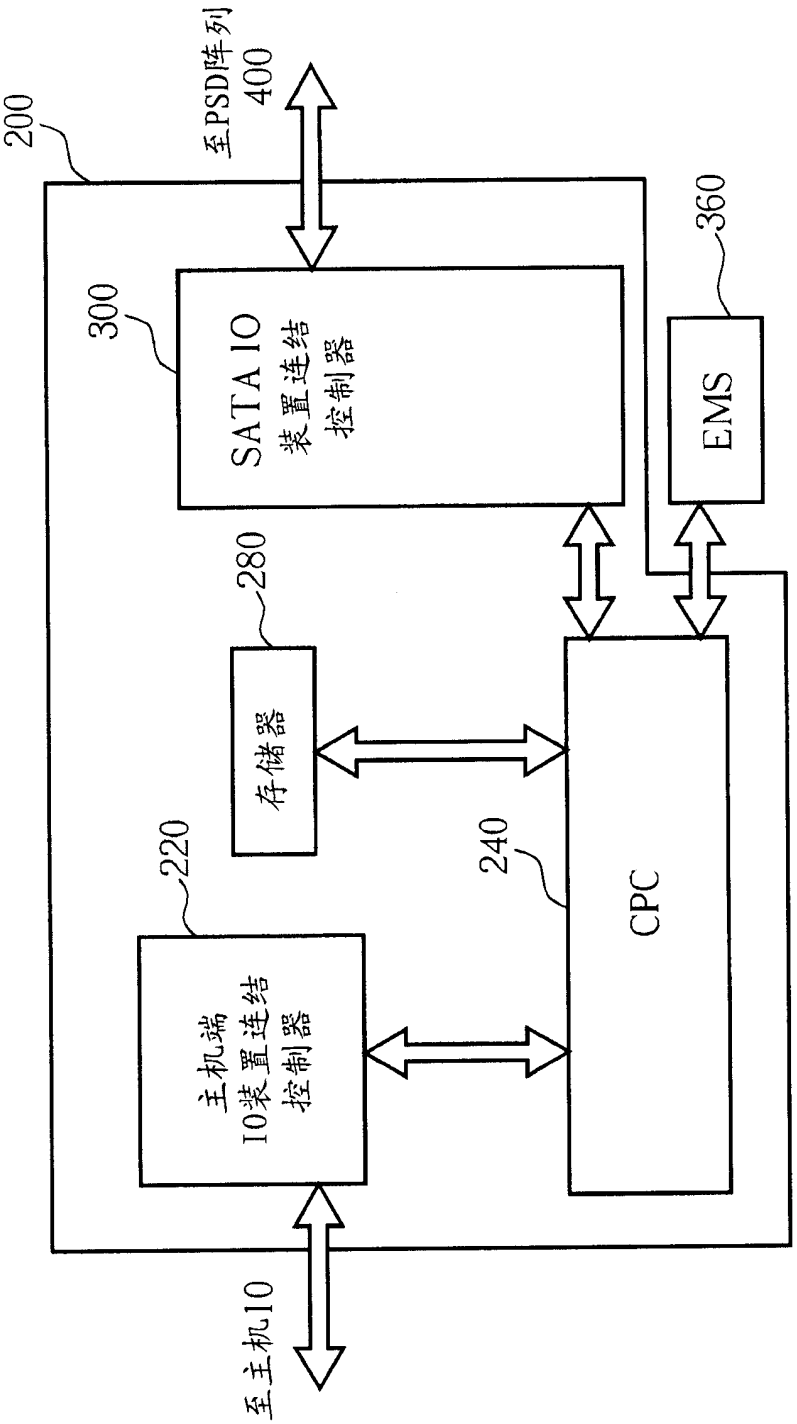

200
至PSD阵列
400
300
SATA IO
装置连结
控制器
360
EMS
280
存储器
240
CPC
220
主机端
IO装置连结
控制器
至主机10

图 4

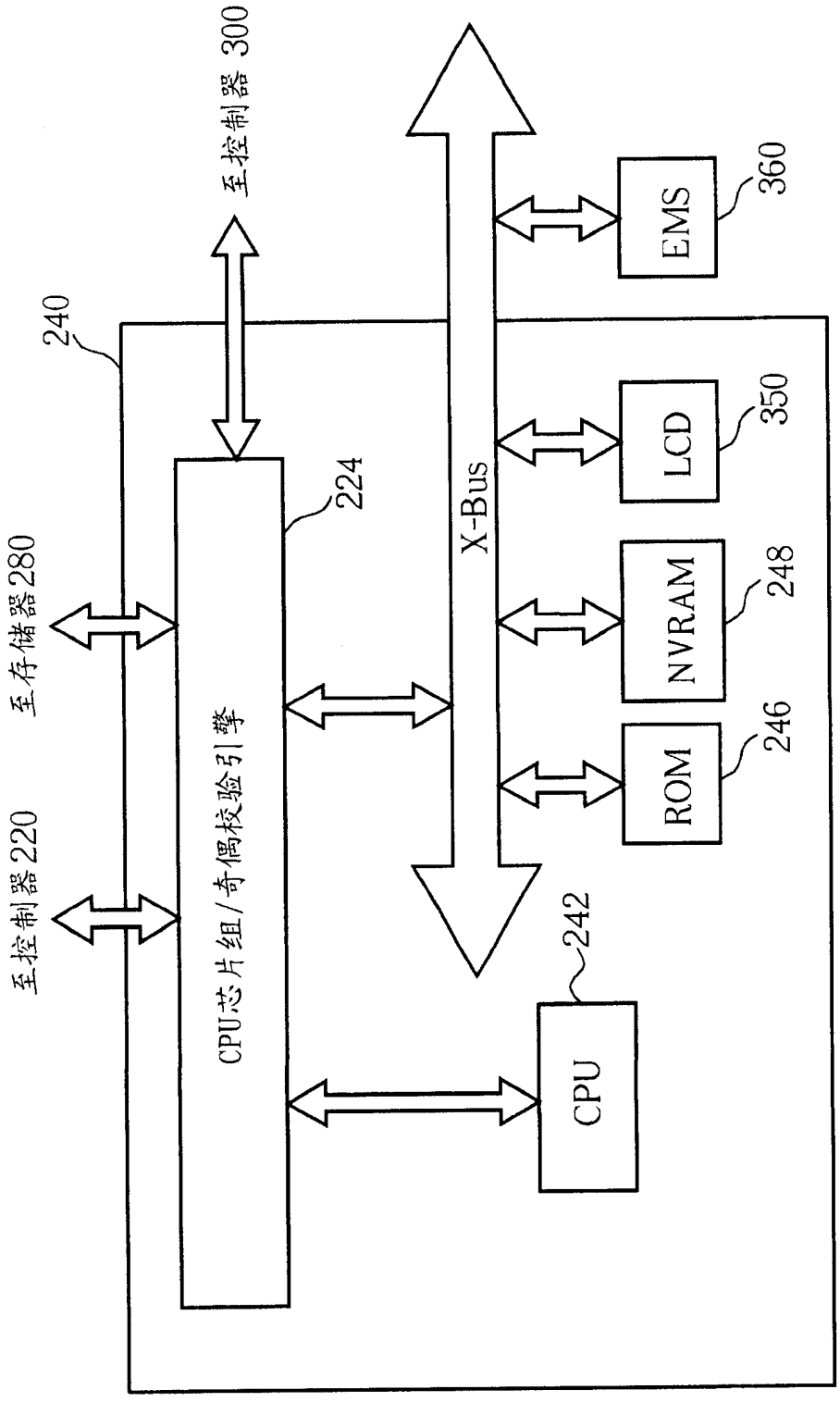

至控制器 300
240
至存储器280
至控制器220
CPU芯片组/奇偶校验引擎
224
X-Bus
EMS
360
LCD
350
NVRAM
248
ROM
246
CPU
242

图 5

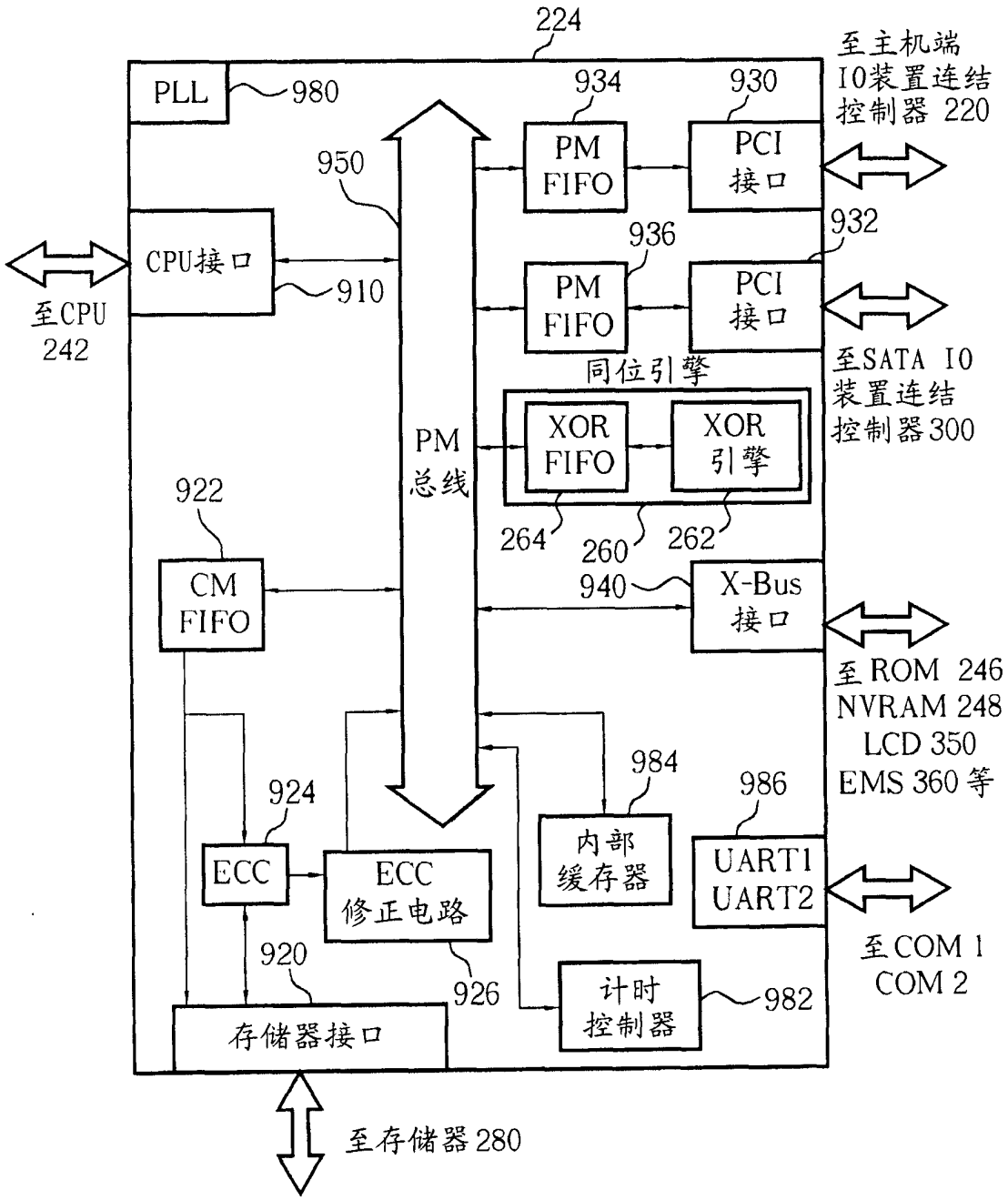

224
至主机端
IO装置连结
控制器 220
PLL
980
934
930
PM
FIFO
PCI
接口
950
936
932
CPU接口
910
至CPU
242
PM
FIFO
PCI
接口
至SATA IO
装置连结
控制器300
同位引擎
XOR
FIFO
XOR
引擎
PM
总线
922
264
260
262
CM
FIFO
940
X-Bus
接口
至ROM 246
NVRAM 248
LCD 350
EMS 360 等
984
986
924
ECC
ECC
修正电路
内部
缓存器
UART1
UART2
至COM 1
COM 2
920
926
计时
控制器
982
存储器接口
至存储器280

图 6

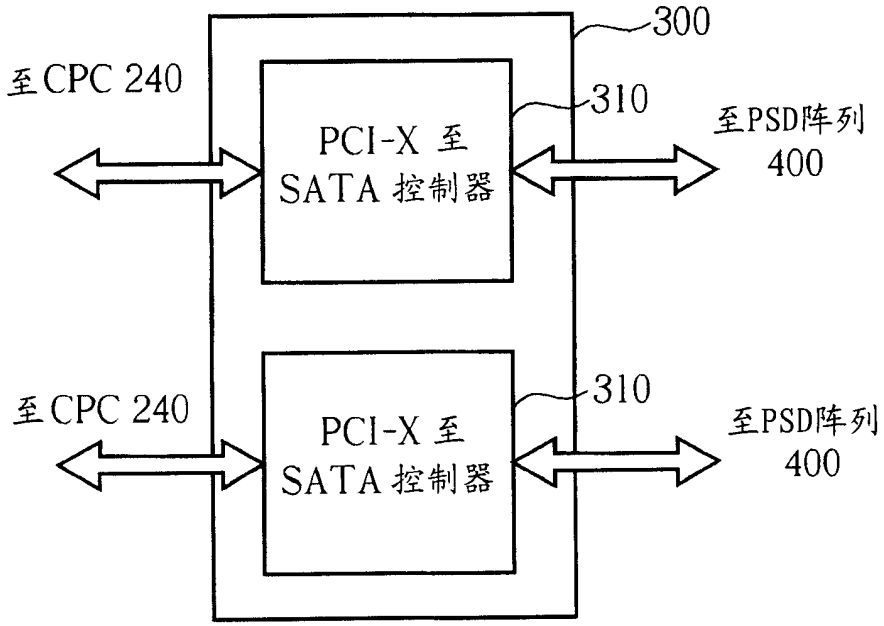

300
至CPC 240
PCI-X 至
SATA 控制器
310
至PSD阵列
400
至CPC 240
PCI-X 至
SATA 控制器
310
至PSD阵列
400

图 7

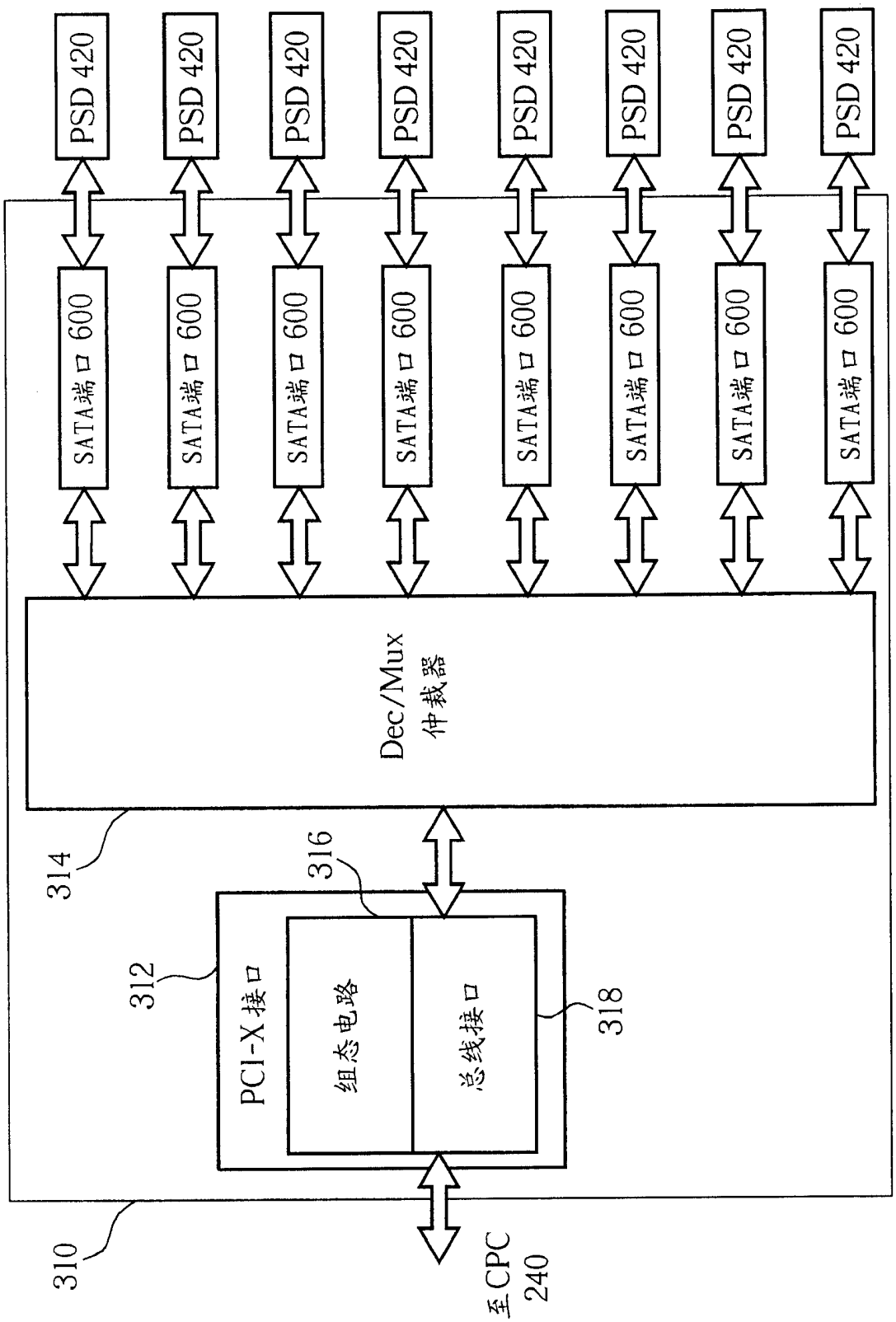

PSD 420
PSD 420
PSD 420
PSD 420
PSD 420
PSD 420
PSD 420
PSD 420
SATA端口 600
SATA端口 600
SATA端口 600
SATA端口 600
SATA端口 600
SATA端口 600
SATA端口 600
SATA端口 600
Dec/Mux
仲裁器
314
316
312
PCI-X接口
组态电路
总线接口
318
310
至CPC
240

图 8

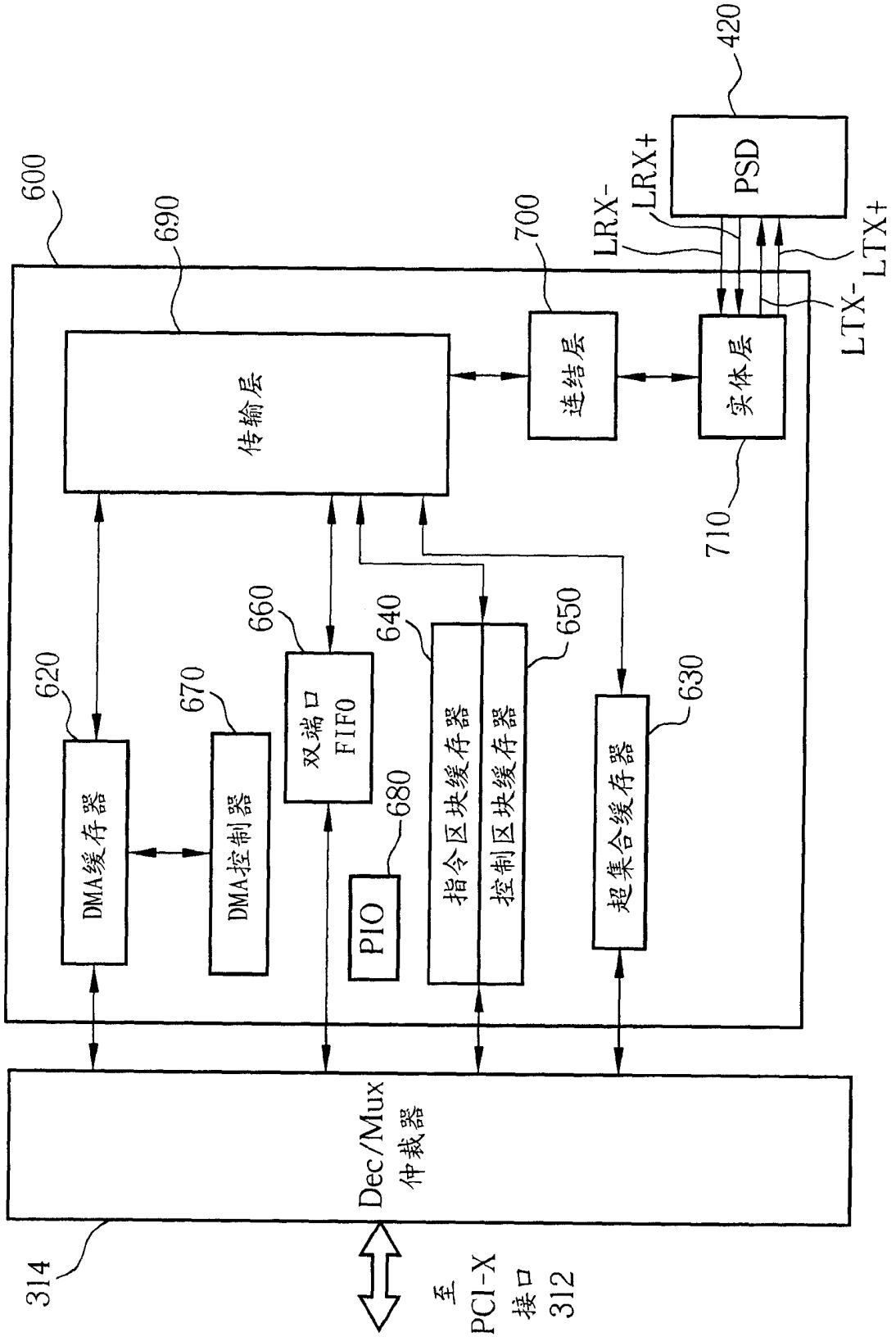

600
690
700
420
PSD
LRX-
LRX+
LTX+
LTX-
传输层
连结层
实体层
710
620
670
660
640
650
630
DMA缓存器
DMA控制器
双端口FIFO
680
PIO
指令区块缓存器
控制区块缓存器
超集合缓存器
Dec/Mux
仲裁器
314
至
PCI-X
接口
312

图 9

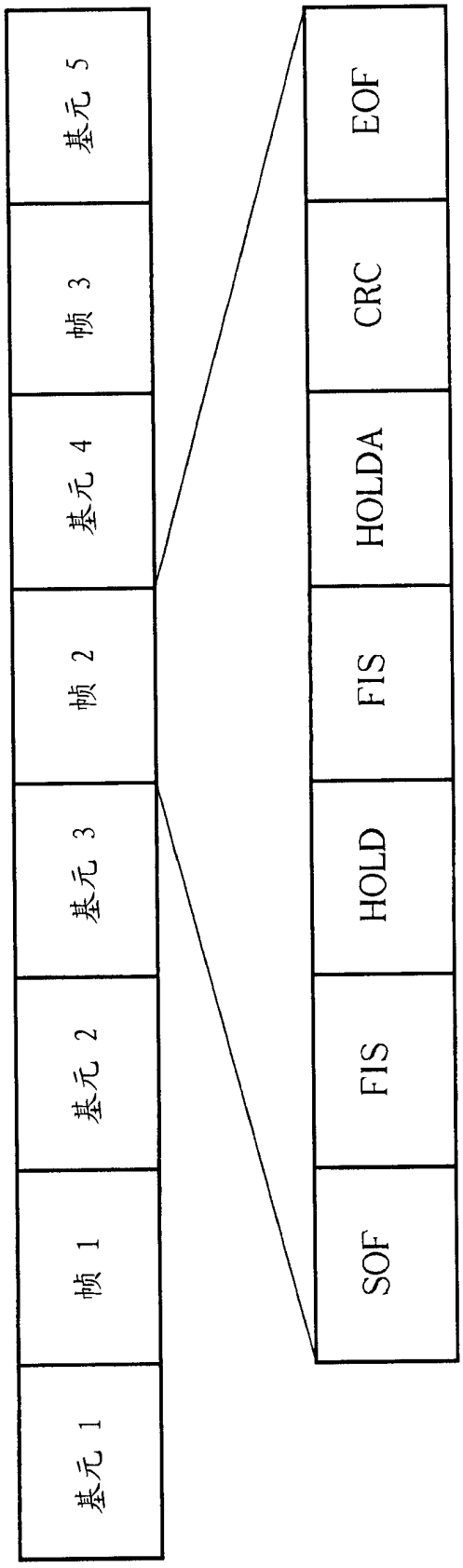


图 10

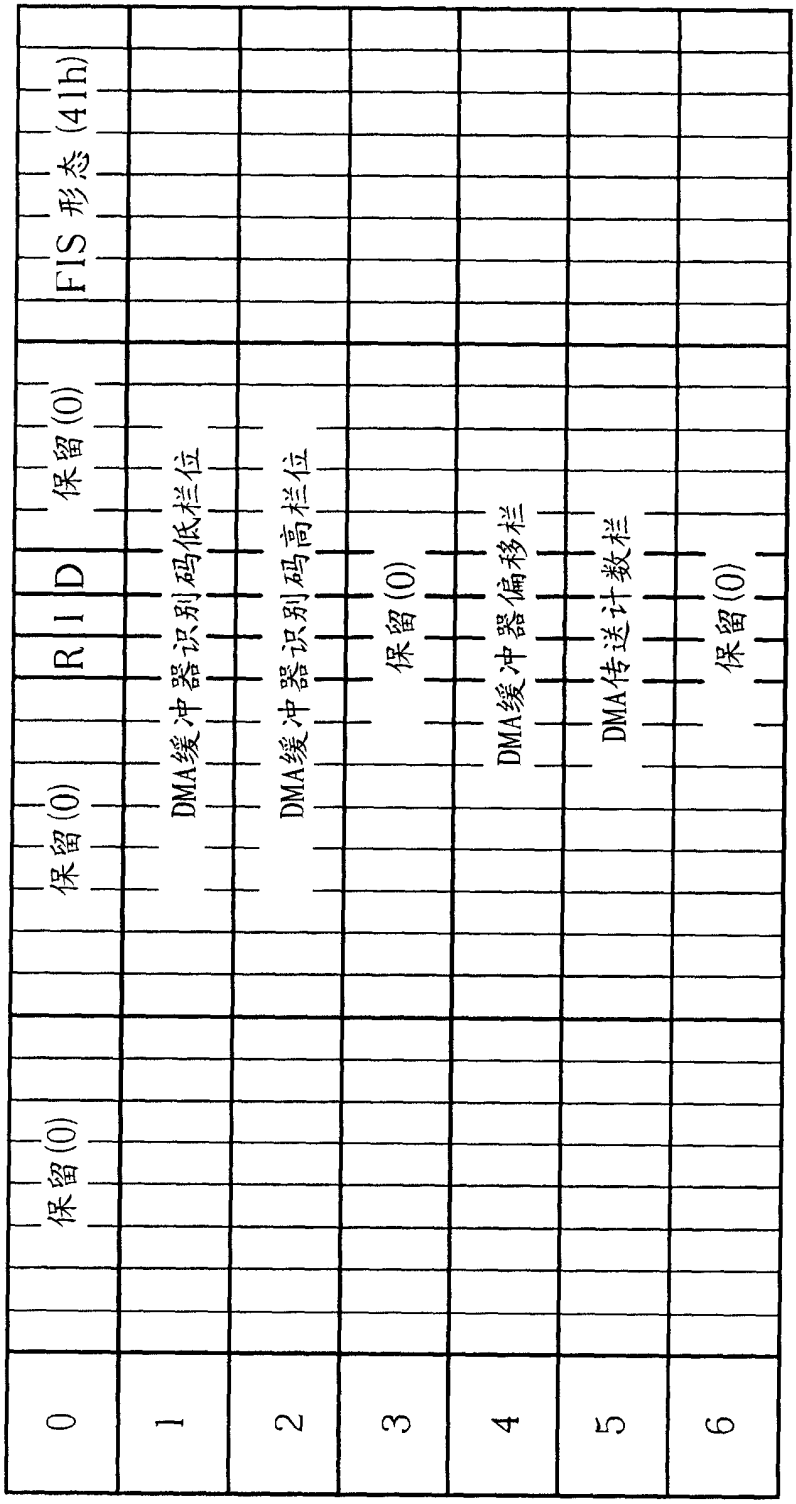

0
保留(0)
保留(0)
R I D
保留(0)
FIS 形态 (41h)
1
DMA缓冲器识别码低栏位
2
DMA缓冲器识别码高栏位
3
保留(0)
4
DMA缓冲器偏移栏
5
DMA传送计数栏
6
保留(0)

图 11

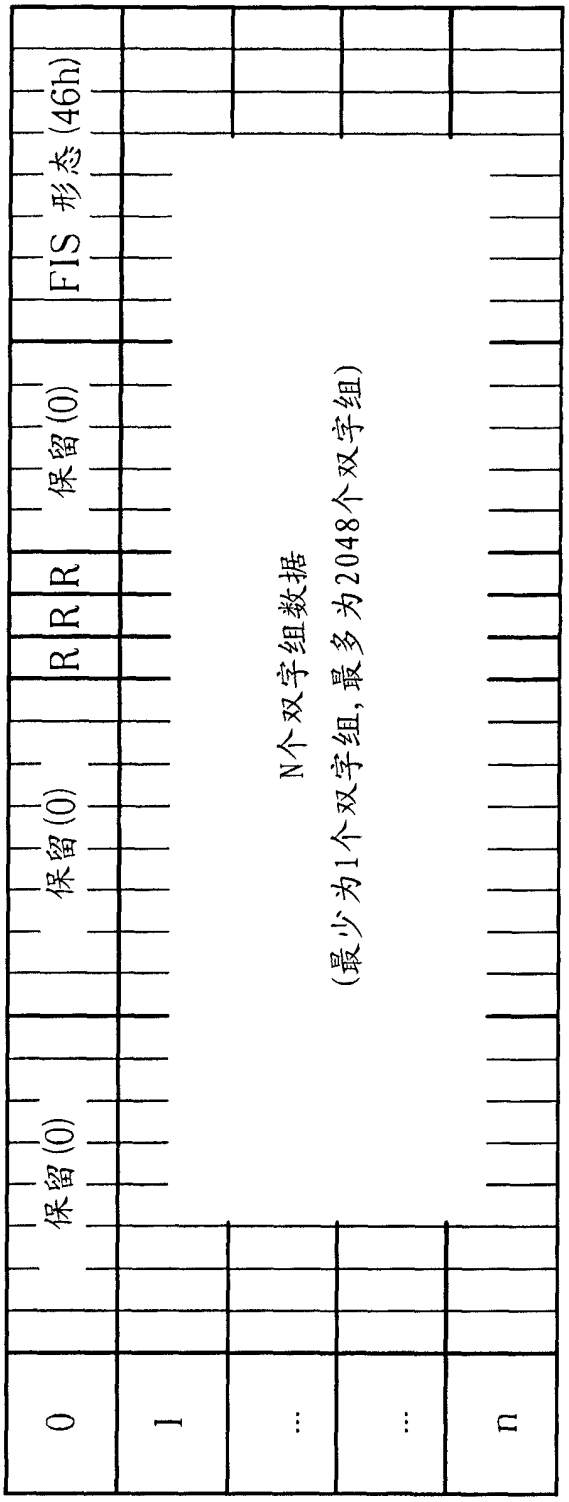


图 12

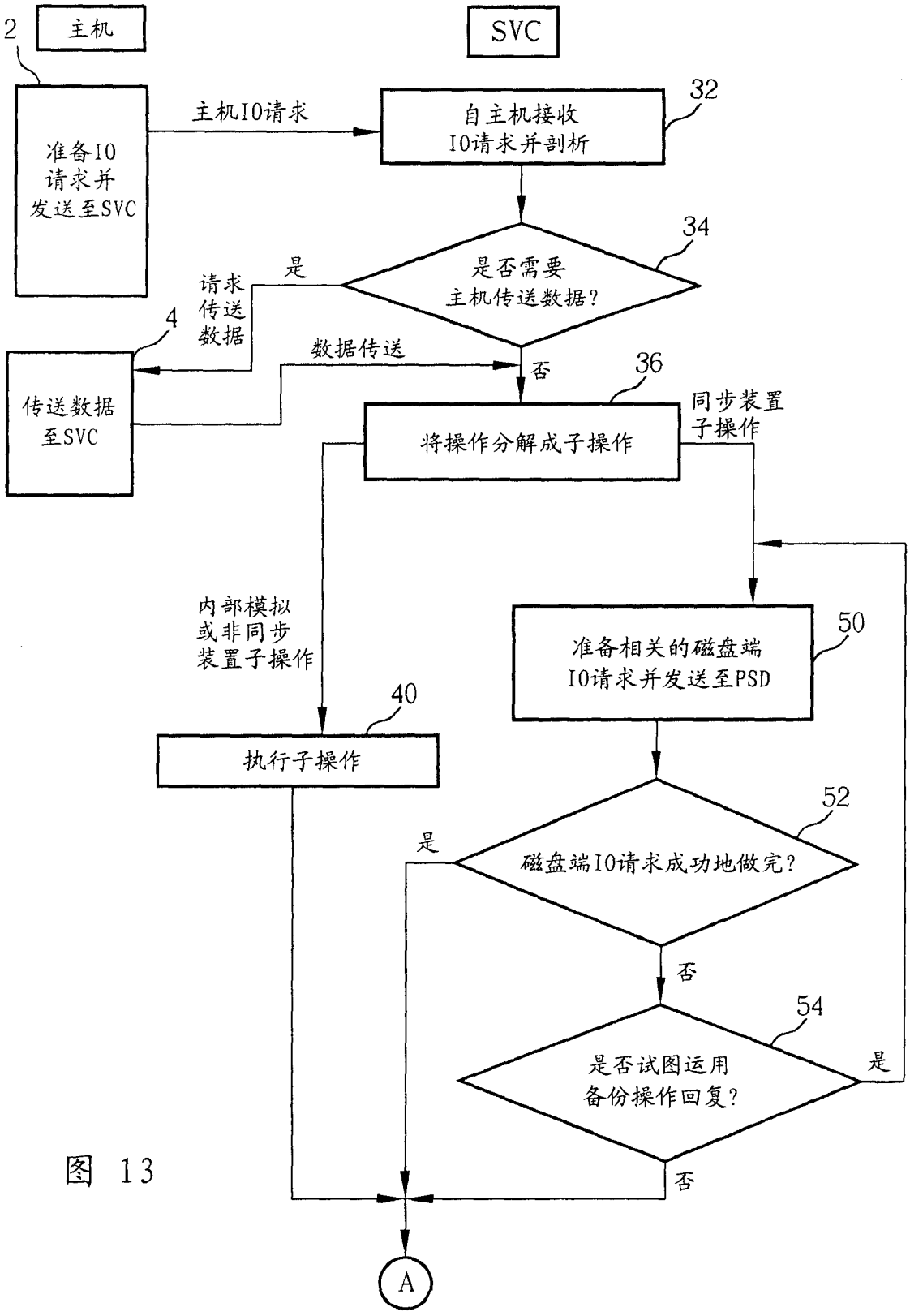

2
主机
SVC
准备IO
请求并
发送至SVC
主机IO请求
自主机接收
IO请求并剖析
32
34
是否需要
主机传送数据?
是
请求
传送
数据
4
传送数据
至SVC
数据传送
否
36
同步装置
子操作
将操作分解成子操作
内部模拟
或非同步
装置子操作
50
准备相关的磁盘端
IO请求并发送至PSD
40
执行子操作
52
是
磁盘端IO请求成功地做完?
否
54
是否试图运用
备份操作回复?
是
否
A

图 13

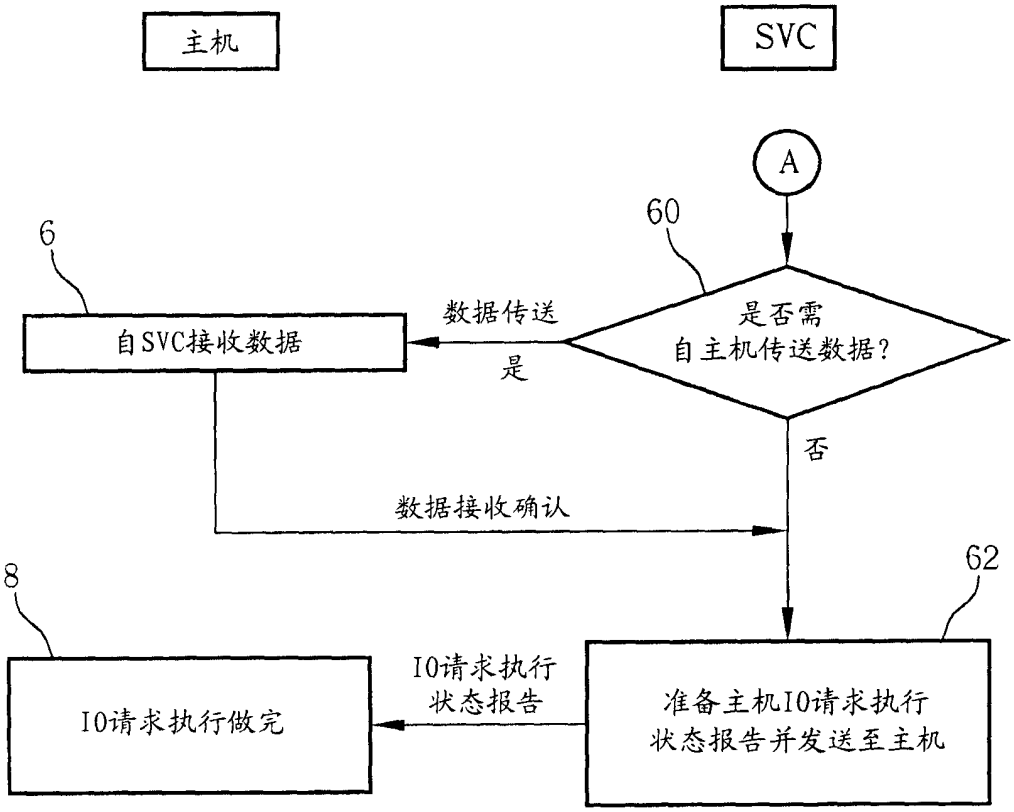

主机
SVC
A
60
6
自SVC接收数据
数据传送
是
是否需
自主机传送数据?
否
数据接收确认
62
8
IO请求执行做完
IO请求执行
状态报告
准备主机IO请求执行
状态报告并发送至主机

图 14

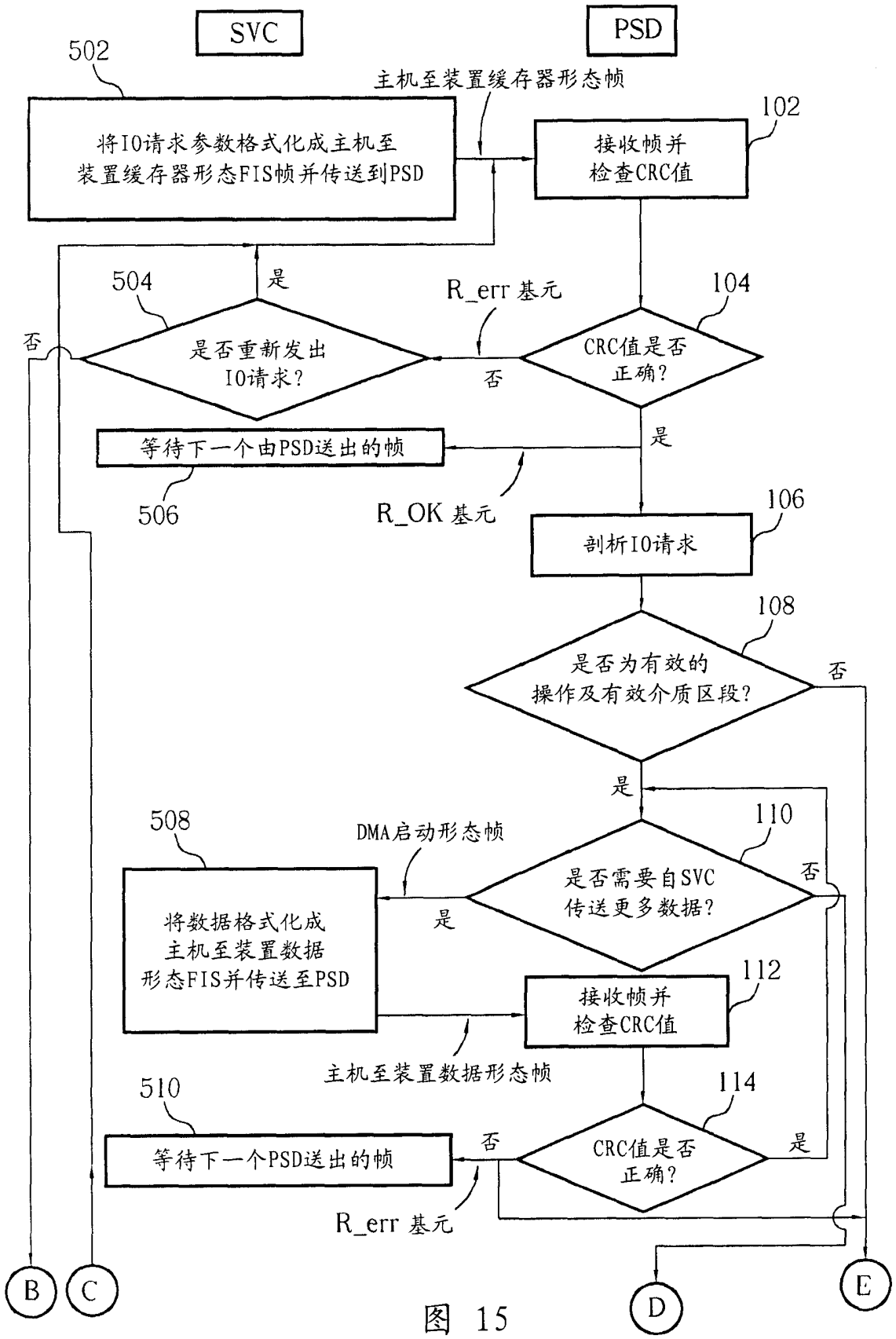

SVC
PSD
502
主机至装置缓存器形态帧
将IO请求参数格式化成主机至装置缓存器形态FIS帧并传送到PSD
接收帧并检查CRC值
102
504
是
R_err 基元
104
否
是否重新发出IO请求?
否
CRC值是否正确?
是
等待下一个由PSD送出的帧
506
R_OK 基元
106
剖析IO请求
108
是否为有效的操作及有效介质区段?
否
是
508
DMA启动形态帧
110
是否需要自SVC传送更多数据?
否
是
将数据格式化成主机至装置数据形态FIS并传送至PSD
112
接收帧并检查CRC值
主机至装置数据形态帧
510
114
等待下一个PSD送出的帧
否
CRC值是否正确?
是
R_err 基元
B
C
D
E

图 15

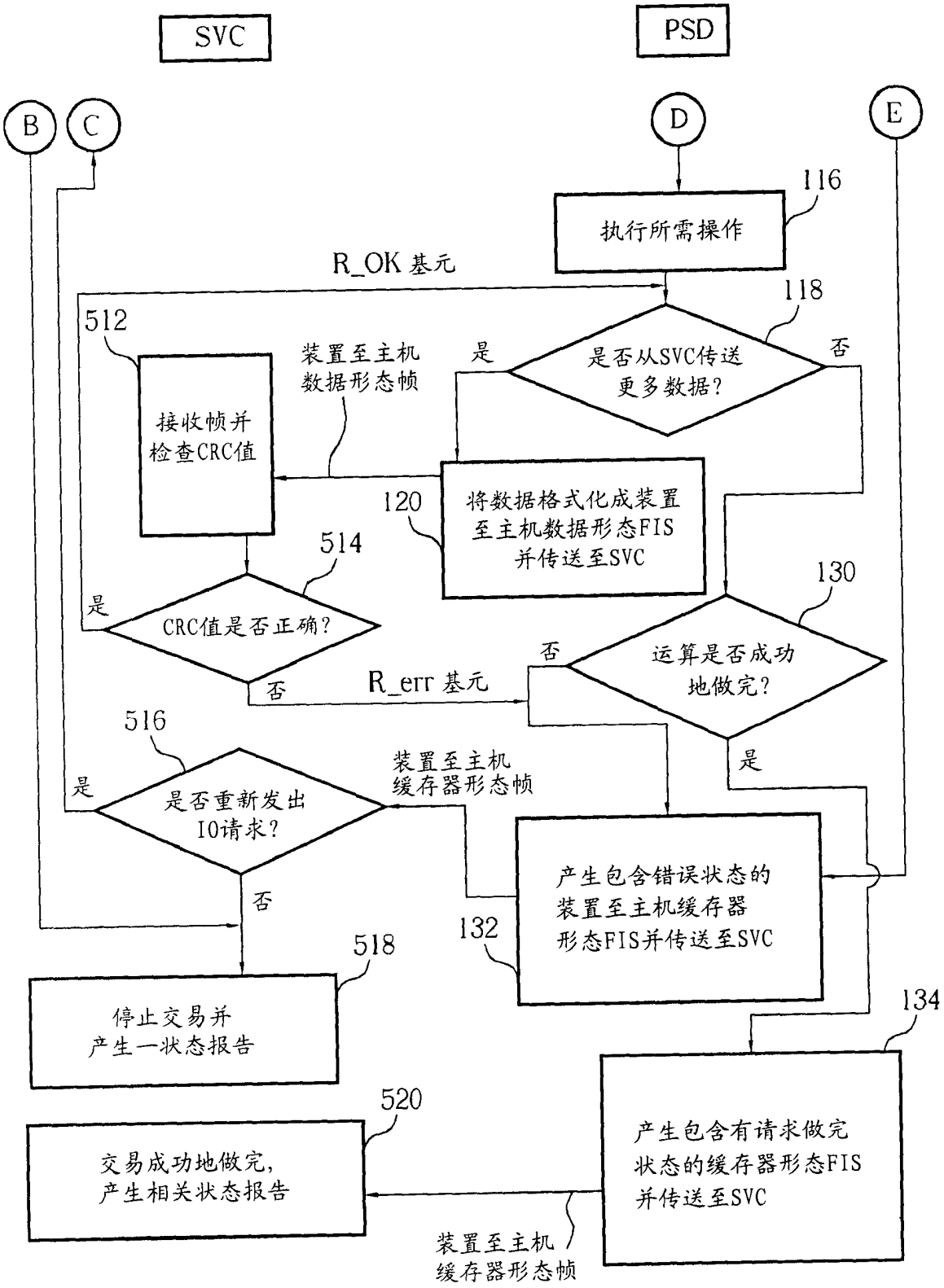

SVC
PSD
B
C
D
E
116
执行所需操作
R_OK 基元
118
512
是否从SVC传送
更多数据?
是
否
装置至主机
数据形态帧
接收帧并
检查CRC值
120
将数据格式化成装置
至主机数据形态FIS
并传送至SVC
514
是
CRC值是否正确?
否
R_err 基元
130
否
运算是否成功
地做完?
是
516
是
是否重新发出
IO请求?
装置至主机
缓存器形态帧
否
产生包含错误状态的
装置至主机缓存器
形态FIS并传送至SVC
132
518
停止交易并
产生一状态报告
134
520
交易成功地做完,
产生相关状态报告
产生包含有请求做完
状态的缓存器形态FIS
并传送至SVC
装置至主机
缓存器形态帧

图 16

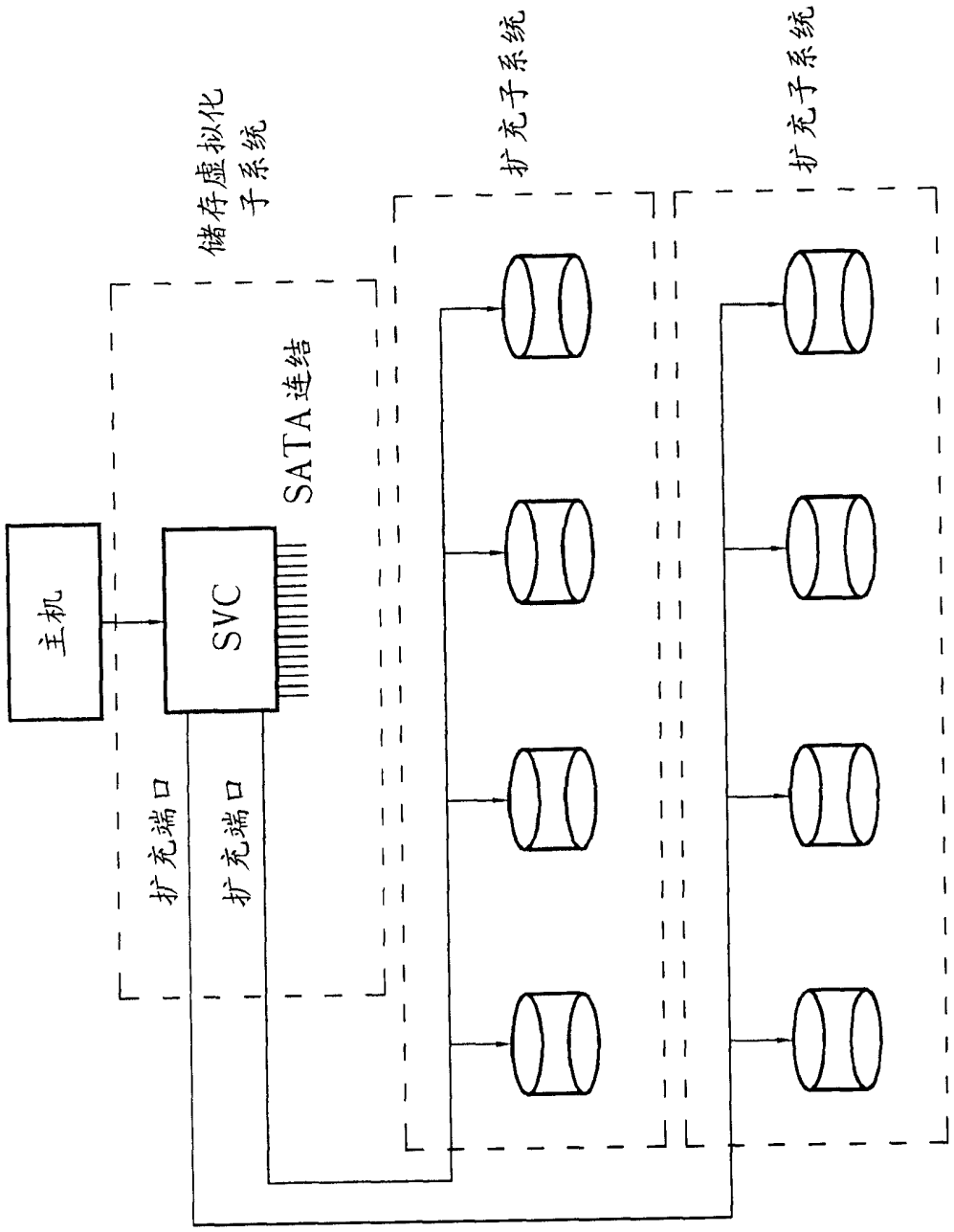


图 17

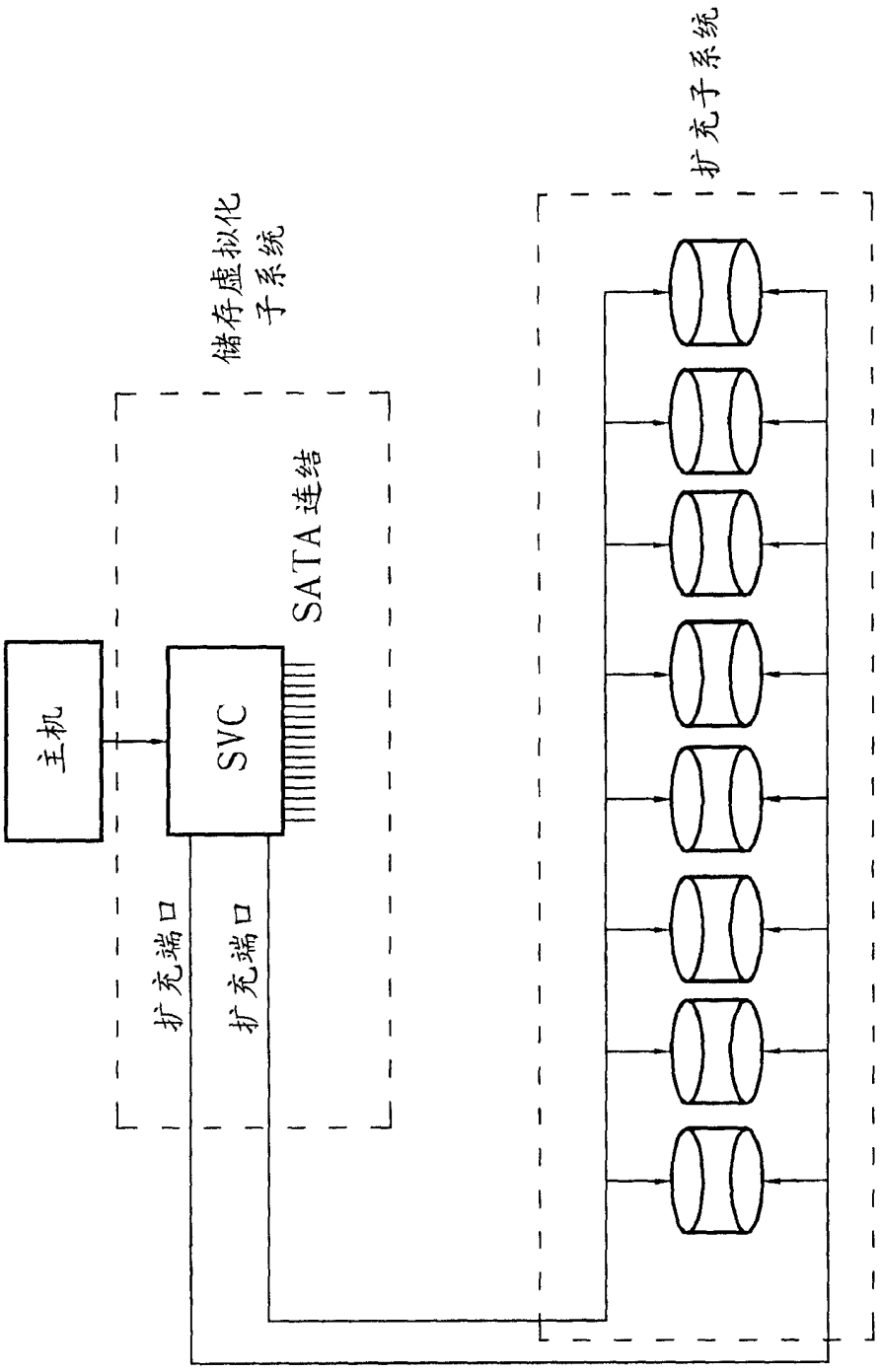


图 18

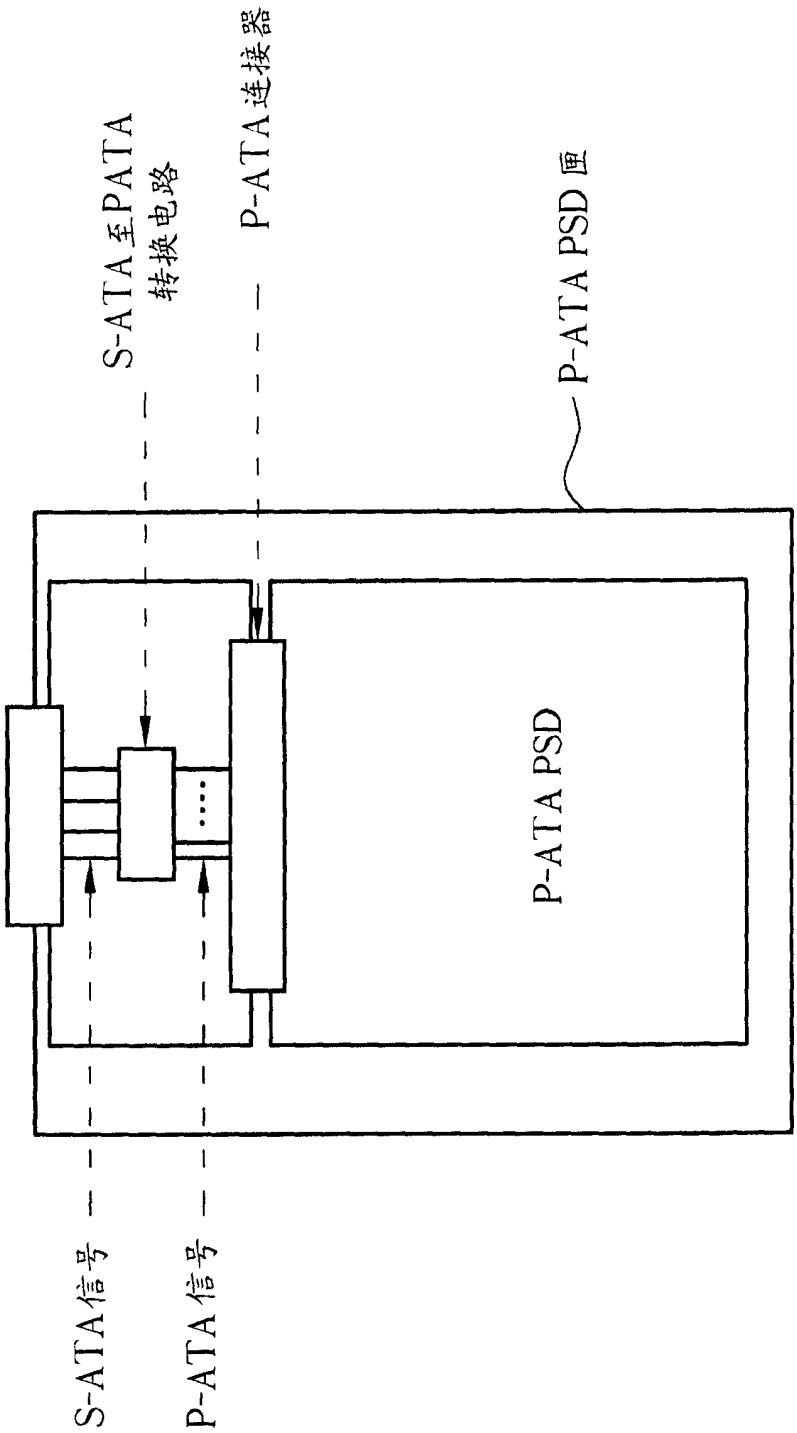


图 19

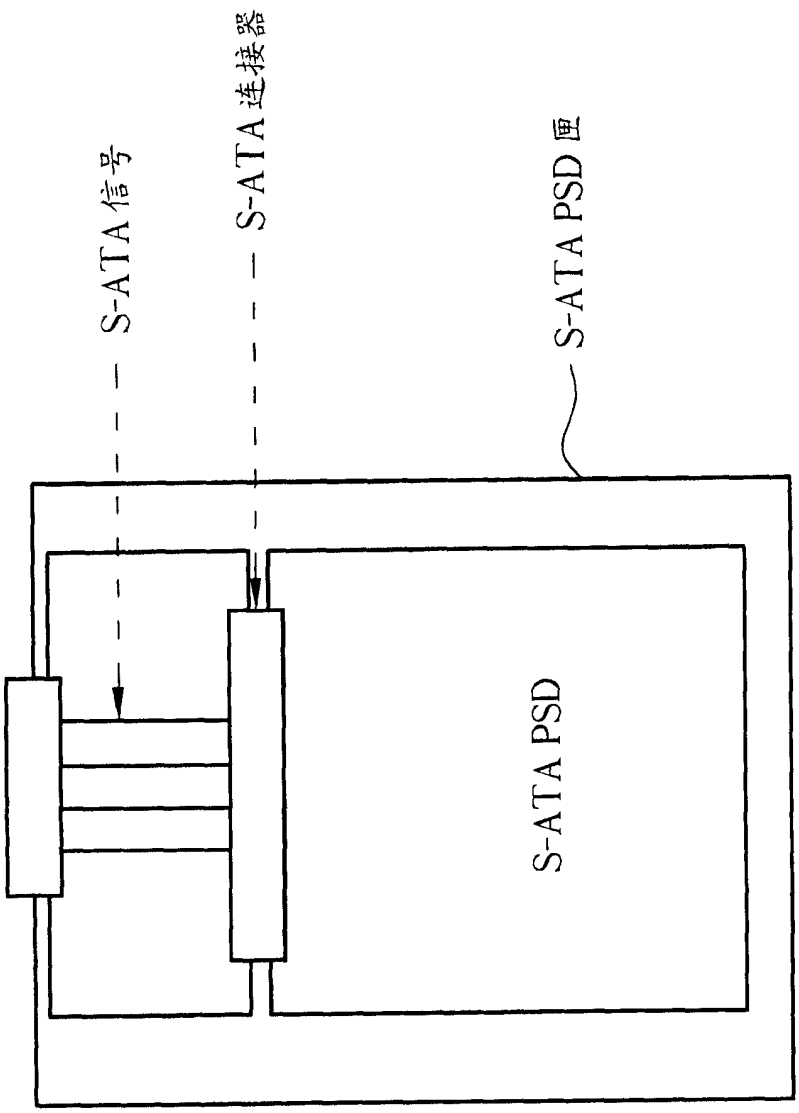


图 20