

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁷

G06F 9/06

[12] 发明专利申请公开说明书

[21] 申请号 98805292.X

[43]公开日 2000年6月21日

[11]公开号 CN 1257591A

[22]申请日 1998.3.19 [21]申请号 98805292.X

[30]优先权

[32]1997.3.20 [33]US[31]08/821,745

[86]国际申请 PCT/US98/05534 1998.3.19

[87]国际公布 WO98/41916 英 1998.9.24

[85]进入国家阶段日期 1999.11.19

[71]申请人 凤凰技术有限公司

地址 美国加利福尼亚州

[72]发明人 Y·拉卡维 I·安德森

A·加斯滕 J·罗奇

M·P·R·伯顿

[74]专利代理机构 中国专利代理(香港)有限公司

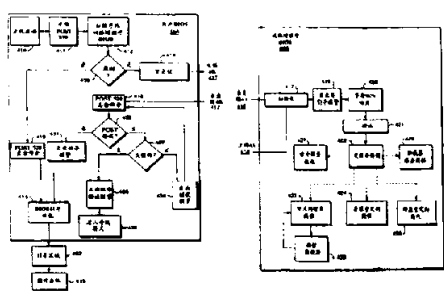
代理人 王 勇 叶恺东

权利要求书 10 页 说明书 21 页 附图页数 10 页

[54]发明名称 远程管理没有一个运行的操作系统的计算机的网络增强型 BIOS

[57]摘要

本发明提供一种在引导计算机操作系统之前或操作系统故障之后通过网络与计算机通信的方法和系统。在网络增强型 BIOS(600) 中实现一个多任务内核。NIC 设备驱动程序中的外部引用被解析为引用由网络增强型 BIOS 提供的服务。可以用与计算机通过网络相连的工作站来在加载操作系统之前或操作系统故障之后访问和设置计算机的状态。多任务内核可以与常规 BIOS (500)同时操作。计算机可进一步配备用于在 POST 故障或操作系统崩溃时向工作站报警的装置。



ISSN 1008-4274

权 利 要 求 书

1. 一种操作计算机的方法，所述计算机包括中央处理单元（CPU）和将所述计算机与网络相连的网络接口，该方法包含：

在 CPU 系统启动时在所述 CPU 上自动执行一个初始启动软件；

5 在完成所述启动软件的至少第一部分时加载并在所述 CPU 上执行网络增强型软件，所述网络增强型软件支持一个与通过所述网络接口连接到所述计算机网络的工作站通信的协议；以及

10 在所述网络增强型软件的所述加载之后加载并在所述 CPU 上执行操作系统软件，所述操作系统软件可被所述 CPU 上执行的用户级应用程序访问。

2. 权利要求 1 的方法，其中，所述网络接口包含网络接口卡（NIC），与所述网络通信的所述协议包含支持所述 NIC 所用驱动程序的网络协议堆栈。

15 3. 权利要求 2 的方法，其中，所述网络协议堆栈支持一个符合 ODI 规范的至少一个子集的网络驱动程序层。

4. 权利要求 2 的方法，其中，所述网络协议堆栈支持一个符合 NDIS 规范的至少一个子集的网络驱动程序层。

20 5. 权利要求 1 的方法，其中，所述网络接口包含通过所述 CPU 相连的调制解调器，与所述网络通信的所述协议支持与所述网络接口的调制解调器接口。

6. 权利要求 1 的方法，进一步包含通过所述网络接口向与所述网络相连的工作站发送 CPU 系统启动的通知的步骤。

7. 权利要求 6 的方法，其中，所述网络接口包含 NIC，及所述通知包含通过所述网络发送 SNMP 报警消息。

25 8. 权利要求 7 的方法，其中所述 SNMP 报警消息进一步包含 CPU 系统复位的原因的标识。

9. 权利要求 6 的方法，其中，所述网络接口包含 NIC，所述方法进一步包含发送指示操作系统已经加载的 SNMP 报警消息。

30 10. 权利要求 1 的方法，其中，没有由所述网络增强型软件建立起与附接到所述网络的远程工作站的网络连接，所述操作系统软件就将不被执行。

11. 权利要求 10 的方法，包含的另外步骤为，通过所述网络从所

述远程工作站下载所述启动软件的第二部分，所述操作系统软件直到所述启动软件的第二部分已经被下载并执行时才执行。

12. 权利要求 1 的方法，其中，所述网络增强型软件以多任务方式执行所述 CPU 上的多个软件线程。

5 13. 权利要求 12 的方法，其中，所述 CPU 以至少两种模式操作，包括实模式和保护模式，所述网络增强型软件在所述 CPU 的所述保护模式下执行所述软件线程中的所述多个。

14. 权利要求 13 的方法，其中，所述启动软件在所述 CPU 的实模式下执行。

10 15. 权利要求 14 的方法，进一步包含的步骤是，在所述 CPU 已经开始执行所述网络增强型软件之后执行所述启动软件的第二部分，所述启动软件和所述网络增强型软件在所述 CPU 已经开始执行所述网络增强型软件之后共享对所述 CPU 的控制。

16. 权利要求 15 的方法，其中，所述网络增强型软件控制所述 CPU 的实模式和保护模式的转换。

17. 权利要求 1 的方法，其中，所述网络增强型软件允许与所述计算机网络相连的远程工作站访问所述计算机。

18. 权利要求 17 的方法，其中，所述网络增强型软件内的多任务线程允许所述远程工作站控制所述 CPU。

20 19. 权利要求 17 的方法，其中，所述网络增强型软件将通常是定向到所述计算机上的视频控制器的输出通过所述网络接口传递到所述远程工作站。

25 20. 权利要求 18 的方法，进一步包含的步骤是，从远程工作站生成一个对所述计算机的所述 CPU 上执行的 RPC 服务器例程的一个远程过程调用（RPC），所述 RPC 服务器例程阻塞其自己的执行，直到所述远程过程调用的结果有数据可用于返回到所述远程工作站。

21. 权利要求 17 的方法，其中，所述网络增强型软件允许所述远程工作站输入模拟在附接到所述计算机的键盘上输入的击键的效果的消息。

30 22. 权利要求 21 的方法，其中，所述消息被所述计算机的所述 CPU 上执行的远程过程调用（RPC）服务器接收，所述 RPC 服务器将所述消息插入与所述计算机相连的键盘控制器。

23. 权利要求 17 的方法，其中，所述远程工作站显示由所述 CPU 上执行的所述启动软件报告的消息和系统错误。

24. 权利要求 1 的方法，进一步包含的步骤是，所述计算机在遇到预定的错误条件时自动进入诊断状态，所述诊断状态阻止所述操作系统软件的加载。

25. 权利要求 17 的方法，进一步包含的步骤是，让所述计算机进入所述操作系统软件将不加载的诊断状态，所述方法进一步包含的步骤是，通过所述计算机网络从所述远程工作站向所述计算机下载另外的软件程序，其中所述另外的软件程序是在所述 CPU 上执行的。

26. 权利要求 17 的方法，其中所述远程工作站可进一步执行所述 CPU 上执行的所述启动软件内的控制开机系统测试 (POST) 例程的步骤。

27. 权利要求 17 的方法，其中，所述远程工作站可迫使所述计算机进入阻止所述操作系统加载的诊断状态，从而所述网络增强型软件可用来进一步与所述远程工作站通信。

28. 权利要求 17 的方法，进一步包含的步骤是，所述远程工作站导致数据被从与所述 CPU 本地连接的直接存取存储器读取，所述数据被进一步从所述计算机传输到所述远程工作站。

29. 权利要求 17 的方法，进一步包含的步骤是，所述远程工作站向所述计算机传输数据，所述数据被写到与所述 CPU 本地连接的直接存取存储器。

30. 权利要求 1 的方法，进一步包含的步骤是，从远程工作站向所述计算机下载所述启动软件或所述网络增强型软件的部分，从而，这种启动软件和网络增强型软件将在下一次系统启动时被执行。

31. 权利要求 1 的方法，进一步包含的步骤是，执行所述计算机上的 RPC 服务器，由此，在与所述网络相连的远程工作站上执行的 RPC 客户机能在所述计算机上执行远程过程调用。

32. 权利要求 1 的方法，其中，所述启动软件包括作为所述启动软件的所述第一部分的一部分的早期 POST 部分和 POST 的一个剩余部分，所述方法进一步包括的步骤为：

在所述 CPU 加载所述网络增强型软件之前执行所述早期 POST；以及

在加载所述网络增强型软件之后执行所述 POST 的剩余部分。

33. 权利要求 1 的方法，进一步包含所述网络增强型软件验证从所述网络接收的任何消息的步骤。

34. 权利要求 33 的方法，其中所述消息验证是用一个秘密密钥进行的。

35. 权利要求 1 的方法，其中所述 CPU 以至少两种模式操作，包括保护模式，所述方法进一步包含的步骤为：

在保护模式下执行所述网络增强型软件和所述操作系统软件；以及

10 加载所述操作系统软件后终止所述网络增强型软件的执行。

36. 权利要求 1 的方法，进一步包含的步骤是，加载所述操作系统软件后在内存中保留所述网络增强型软件的一部分。

37. 一种操作计算机的方法，所述计算机包括中央处理单元 (CPU) 和将所述计算机与网络相连的网络接口，所述 CPU 在包含实模式与保护模式的至少两种模式中操作，所述方法包含：

以实模式在所述 CPU 上执行第一基本输入/输出系统 (BIOS)，所述第一 BIOS 进行所述计算机的开机系统测试；

在所述第一 BIOS 的一部分的执行之后，以保护模式在所述 CPU 上执行第二 BIOS，所述第二 BIOS 支持网络协议堆栈；以及

20 在所述第二 BIOS 的一部分的执行之后，以保护模式在所述 CPU 上执行操作系统，所述操作系统软件支持在所述 CPU 上执行的用户级应用软件程序。

38. 一个能够与连接计算机网络的工作站通信的网络增强型计算机，所述计算机包含：

25 中央处理单元 (CPU)

与所述 CPU 相连的网络接口，所述网络接口将所述计算机与所述计算机网络相连；

与所述 CPU 相连的非易失性存储器；

30 初始启动软件程序，在 CPU 系统启动时所述 CPU 自动执行所述启动软件，所述启动软件的至少第一部分存储在所述非易失性存储器中；

网络增强型软件，当所述启动软件的至少第一部分完成时所述网

网络增强型软件在所述 CPU 上被执行, 所述网络增强型软件支持与通过所述网络接口连接到所述计算机网络的所述工作站通信的协议; 以及

在所述 CPU 上执行的操作系统软件, 所述操作系统软件可被用户级应用程序访问, 在所述网络增强型软件的加载之后所述操作系统软件在所述 CPU 上被加载。

39. 权利要求 38 的网络增强型计算机, 其中所述网络接口是包含可被所述 CPU 访问的设备驱动程序的网络接口卡 (NIC), 与所述网络通信的所述协议包含一个与所述设备驱动程序接口的网络协议堆栈。

40. 权利要求 39 的网络增强型计算机, 其中设备驱动程序实质上符合 ODI 驱动程序规范, 所述网络协议堆栈提供 ODI 链路支持层协议的至少一个子集来与所述设备驱动程序接口。

41. 权利要求 39 的网络增强型计算机, 其中设备驱动程序实质上符合 NDIS 驱动程序规范, 所述网络协议堆栈提供 NDIS 协议的至少第二个子集来与所述设备驱动程序接口。

42. 权利要求 38 的网络增强型计算机, 其中, 所述网络接口是与所述 CPU 相连的调制解调器, 与所述网络通信的所述协议支持连接与所述网络接口的调制解调器接口。

43. 权利要求 38 的网络增强型计算机, 进一步包含一个程序段, 用于通过所述网络接口向与所述网络相连的工作站发送 CPU 系统启动的通知, 所述通知在所述操作系统的所述执行之前发送。

44. 权利要求 43 的网络增强型计算机, 其中所述网络接口是 NIC, 所述通知包含 SNMP 报警消息。

45. 权利要求 38 的网络增强型计算机, 进一步包含一种在由所述网络增强型软件建立起与附接到所述网络的远程工作站的网络连接之前阻止所述操作系统软件的执行的装置。

46. 权利要求 38 的网络增强型计算机, 进一步包含一种用于通过所述网络从所述远程工作站下载所述启动软件的所述第二部分的装置, 所述操作软件直到所述启动软件的所述第二部分已经被下载时才执行。

47. 权利要求 38 的网络增强型计算机, 其中, 所述网络增强型软件包含:

多任务内核；以及

在所述 CPU 上所述多任务内核内执行的多个软件线程。

48. 权利要求 47 的网络增强型计算机，其中，所述 CPU 以包括实模式和保护模式的至少两种模式操作，所述多任务内核包括在所述 CPU 的保护模式下操作的内核。

49. 权利要求 48 的网络增强型计算机，其中，所述启动软件在所述 CPU 的实模式下执行。

50. 权利要求 49 的网络增强型计算机，其中所述启动软件和所述多任务内核在所述 CPU 开始执行所述网络增强型软件后共享对所述 CPU 的控制。

51. 一个能够与计算机网络交互作用的计算机，所述计算机包含：提供包括实模式和保护模式的至少两种模式的中央处理单元（CPU）；

将所述计算机与所述计算机网络相连的网络接口；

与所述 CPU 相连的非易失性存储器；

第一基本输入/输出系统（BIOS），所述第一 BIOS 至少有一部分存储在所述非易失性存储器中，使得所述 CPU 在系统复位时能调用所述第一 BIOS，所述第一 BIOS 进一步进行所述计算机的开机系统测试；

在所述第一 BIOS 的一部分执行之后以保护模式在所述 CPU 上执行的第二 BIOS，所述第二 BIOS 支持网络协议堆栈；以及

在所述第二 BIOS 的一部分执行之后以保护模式在所述 CPU 上执行的操作系统，所述操作系统支持在所述 CPU 上执行的用户级应用程序。

52. 一个远程管理至少一个计算机的系统，包含：

计算机网络；

与所述计算机网络相连的工作站，所述工作站执行应用软件；以及

计算机，含有：

将所述计算机与所述计算机网络连接的网络接口；

中央处理单元（CPU）；

初始启动软件程序，所述启动软件在 CPU 系统复位时自动执

行;

网络增强型软件,配置成在所述至少第一部分的所述启动软件执行后在所述 CPU 上被执行,所述网络增强型软件支持用于通过所述网络接口与所述工作站通信的协议; 以及

5 在所述 CPU 上执行的操作系统软件,所述操作系统软件可被用户级应用软件程序访问,所述操作系统软件在所述 CPU 上被加载。

53. 权利要求 52 的系统,进一步包含一种用于所述远程工作站访问和控制所述 CPU 的装置。

10 54. 权利要求 53 的系统,进一步包含一种用于所述远程工作站输入模拟在附接到所述计算机的键盘上输入的击键的效果的消息的装置。

15 55. 权利要求 53 的系统,其中,所述计算机可以进入阻止所述操作系统软件加载的诊断状态,所述系统进一步包括一种用于从所述远程工作站向所述计算机下载另外的软件程序的装置,其中所述另外的软件程序在所述 CPU 上执行。

56. 权利要求 53 的系统,其中,所述远程工作站包含一种用于控制所述 CPU 上执行的所述启动软件内开机系统测试 (POST) 例程的装置。

20 57. 一种用于具有系统管理 RAM (SMRAM) 和定期对事件计数器作第一调整的操作系统的计算机系统的故障恢复方法,该方法包含的步骤为:

以固定间隔生成系统管理中断 (SMI);

响应所述 SMI 执行中断服务例程,所述中断服务例程对所述事件计数器作第二调整; 以及

25 响应所述事件计数器达到预定极限时收集系统状态信息。

58. 权利要求 57 的故障恢复方法,进一步包含的步骤是,将所述所收集信息传输到与所述计算机系统通过网络相连的远程工作站。

30 59. 权利要求 58 的故障恢复方法,其中,所收集信息的传输,是由配备用于与所述计算机系统通过网络相连的工作站通信的协议的保护模式内核进行的。

60. 一种能在计算机系统发生故障时收集数据的计算机,所述计算机包含:

中央处理单元 (CPU) ;

与所述 CPU 相连的随机存取存储器, 所述随机存取存储器存储计算机系统状态数据;

5 在所述 CPU 上执行的操作系统, 所述操作系统可被所述 CPU 上执行的 5 用户级应用软件访问;

与所述 CPU 相连的系统管理 RAM (SMRAM) ;

配置成能检测所述计算机系统故障的崩溃检测机构; 以及

10 在所述 SMRAM 中存储的数据收集例程, 所述数据收集例程被配置成根据由所述崩溃检测机构对所述计算机系统故障的检测而在所述 CPU 上执行, 所述数据收集例程存储所述计算机系统状态数据的至少 10 第一部分。

61. 权利要求 60 的计算机, 其中, 所述随机存取存储器包含物理地位于视频控制器上的存储器, 所述计算机系统状态数据包含视频控制器寄存器数据。

15 62. 一种能从连接到计算机网络上的工作站的远程直接存取存储器引导其操作系统的计算机, 所述计算机包含:

中央处理单元 (CPU), 所述 CPU 为请求数据访问提供 BIOS 磁盘服务中断;

非易失性存储器;

20 将所述计算机与所述计算机网络连接的网络接口;

启动软件, 所述启动软件至少有一部分存储在所述非易失性存储器中, 所述启动软件在所述计算机内配置, 使得所述 CPU 能在系统启动时调用所述启动软件, 所述启动软件包含一个用所述 BIOS 磁盘服务中断来引导所述操作系统的操作系统引导程序; 以及

25 网络增强型软件内核, 在执行至少一部分所述启动软件之后及执行所述操作系统引导程序之前至少有一部分所述内核被所述 CPU 执行, 所述内核支持用于与所述工作站通信的协议堆栈, 所述内核重定向所述 BIOS 磁盘服务中断以捕获所述 BIOS 磁盘服务中断并从所述工作站访问所述请求的数据。

30 63. 一种能从连接到计算机网络上的工作站引导其操作系统的计算机, 所述计算机包含:

中央处理单元 (CPU), 所述 CPU 为请求数据访问提供 BIOS 磁盘

服务中断;

非易失性存储器;

将所述计算机与所述计算机网络连接的网络接口卡 (NIC);

5 启动软件, 所述启动软件至少有一部分存储在所述非易失性存储器中, 所述启动软件在所述计算机内配置, 使得所述 CPU 能在系统启动时调用所述启动软件, 所述启动软件包含一个用所述 BIOS 磁盘服务中断来引导所述操作系统的操作系统引导程序; 以及

支持用于通过所述 NIC 与所述工作站通信的协议堆栈的网络多任务内核, 所述多任务内核在所述操作系统引导程序的执行之前被执行, 所述多任务内核执行通过所述 NIC 从所述工作站访问数据的第一
10 磁盘重定向线程, 所述 BIOS 磁盘服务中断被重定向到所述磁盘重定向线程,

由此, 所述操作系统被通过所述磁盘重定向线程从所述工作站引导。

15 64. 一种在具有提供实模式和保护模式的 CPU 的计算机上同时执行实模式操作系统和保护模式内核的方法, 所述计算机进一步提供计时器中断, 所述方法包含:

(a) 在被执行的转移程序保存实模式操作系统的当前状态, 将 CPU 的控制传送给恢复保护模式内核的状态并将所述 CPU 转换到保护
20 模式的部分代码时, 将所述计时器中断引向在实模式下执行的所述转移程序;

(b) 执行部分所述保护模式内核;

(c) 将所述 CPU 转换到实模式并保存保护模式内核的当前状态;

(d) 将所述实模式操作系统恢复到所保存的状态; 以及

25 (e) 执行所述实模式操作系统, 直到被所述计时器中断所中断, 由此所述实模式操作系统将执行, 直到被所述计时器中断所中断, 此时所述保护模式内核将被恢复到所保存的状态并被给予 CPU 控制。

30 65. 权利要求 64 的同时执行实模式操作系统和保护模式内核的方法, 其中, 所述保护模式内核是多任务内核。

66. 权利要求 65 的同时执行实模式操作系统和保护模式内核的方法, 其中, 所述保护模式内核是执行线程的 32 位多任务内核, 当所

述保护模式内核上没有线程执行时,所述保护模式内核将所述 CPU 的控制传送给所述实模式操作系统。

67. 权利要求 3 的方法,其中,所述网络协议堆栈是由在非易失性存储器中存储的软件实现的。

5 68. 权利要求 4 的方法,其中,所述网络协议堆栈是由在非易失性存储器中存储的软件实现的。

69. 一种操作计算机的方法,所述计算机包括中央处理单元(CPU),所述 CPU 以包括实模式和保护模式的至少两种模式操作,所述方法包含:

10 加载并在实模式下的所述 CPU 上执行第一基本输入/输出系统(BIOS),所述第一 BIOS 有个初始部分和一个后继部分,所述第一 BIOS 进行所述计算机的开机系统测试;

在所述第一 BIOS 的所述初始部分的执行之后,加载并在保护模式的所述 CPU 上执行第二 BIOS;

15 所述第二 BIOS 和所述第一 BIOS 的所述后继部分交替地在所述 CPU 上被执行,使得所述第二 BIOS 和所述第一 BIOS 的后继部分看起来在同时执行。

70. 权利要求 69 的方法,其中,所述计算机进一步提供计时器中断,所述同时执行所述第二 BIOS 和所述第一 BIOS 的所述后继部分的步骤包含:

(a) 在被执行的转移程序保存实模式第一 BIOS 的当前状态,将 CPU 的控制传送给恢复保护模式第二 BIOS 的状态并将所述 CPU 转换到保护模式的部分代码时,将所述计时器中断引向在实模式下执行的转移程序;

25 (b) 执行所述保护模式第二 BIOS 的一部分;

(c) 将所述 CPU 转换到实模式并保存保护模式第二 BIOS 的当前状态;

(d) 将所述实模式第一 BIOS 恢复到所保存的状态; 以及

(e) 执行所述实模式第一 BIOS,直到被所述计时器中断所中断,

30 由此,所述实模式第一 BIOS 将执行,直到被所述计时器中断所中断,此时所述保护模式第二 BIOS 将被恢复到所保存的状态并被给予 CPU 控制。

说明书

远程管理没有一个运行的操作系统的计算机的网络增强型 BIOS

5 本发明总的来说涉及一种用于在没有计算机上运行的操作系统的
辅助时能通过网络远程管理计算机的方法和系统，更具体来说，涉及
一种用于增强计算机的 BIOS 以便能在没有计算机上执行的操作系统的
辅助时进行远程访问和维护的方法和系统。

导致操作系统故障的系统错误依然是计算机业的一个广泛问题。
这种错误会因为硬件故障、用户错误或其它原因而发生。这些故障—
10 特别是在连网的台式计算机或网络系统服务器中，会导致计算机资源
不能使用的范围扩大和用户停机带来的严重经济损失。

当前，没有有效的处理能彻底消除这种导致操作系统崩溃的系统
错误。在许多情况下，避免再次崩溃的唯一方法是在重新引导操作系
统之前进行对引起崩溃的操作的故障后诊断。然而，由于访问和分析
15 故障计算机的物理困难，及由于使系统可用时间通常极其重要，许多
用户不分析导致崩溃的问题就重新引导操作系统。

需要进行这种访问的一例情形是网络服务器。这种网络服务器对
于组织的效率通常至关重要，而又可被配置成没有进行故障后分析所
必需的某些硬件，诸如键盘和计算机显示器。因此网络操作员经常不
20 对问题进行适当的诊断就匆匆忙忙重新引导网络服务器。

远程分析和管理工作的问题也发生在要求系统管理员远程维护
数台计算机的宽域网或局域网中。在典型的操作中，在远程计算机上
执行的操作系统，允许系统管理员访问和修改远程计算机上的各种参
数。然而，如果发生操作系统崩溃，当前的系统不提供管理员访问或
25 诊断远程计算机的手段。此外，当前的系统通常不允许在远程计算机
上加载操作系统之前访问远程计算机。例如，Burckhartr 等人的美国
专利 5,390,324 号（“Burckhaart 专利”）要求的一种故障恢复
系统，一旦故障计算机装入了计算机的硬盘上二级分区中存储的精简
操作系统，就允许对故障计算机的拨号访问。Burckhaart 专利的系
30 统在检测装置检测到表示一级操作系统故障的系统暂时时，就引导含
有二级操作系统的二级分区。

以下背景描述了 IBM 兼容的个人计算机（PC）的典型结构和启动

过程。然而，这些概念一般适用于各种计算机系统。当系统复位时，CPU 控制被传送给计算机的基本输入输出系统（BIOS）的一部分一称为电源接通系统测试或电源接通自测试（POST）。本文中使用的术语系统复位和系统启动等同于并包括导致计算机的初始程序加载操作的初始化或重新初始化的开始的任何系统启动、重新启动、系统复位或其它操作。

POST 通常存储在只读存储器（ROM）中，用于初始化标准系统部件，诸如系统定时器、系统 DMA（直接存储器存取）控制器、系统存储器控制器、系统 I/O 设备和视频硬件。作为其初始化例程的一部分，POST 为中断向量表设置缺省值。这些缺省值指向 ROM BIOS 中的标准中断处理程序，但是可修改为访问定制的中断处理程序。POST 也进行可靠性测试来检查系统硬件诸如存储器和系统定时器的运行正常。系统初始化和诊断之后，POST 调查系统位于系统中可选硬件卡（适配器）上非易失性存储器上的固件。这是通过扫描特定地址空间寻找具有给定标记的存储器而执行的。如果找到标记，就将控制传递给固件，后者然后就初始化其位于的设备。

进行硬件初始化之后，POST 从引导设备的预定位置—通常是硬盘或软盘驱动器—将一个数据块读入存储器并将控制传递给该数据块中的程序。这个程序称为引导程序加载器，它将一个更大的程序装入存储器。如果该更大程序正确地装入存储器，引导程序就将控制传送给它。操作系统然后被初始化并获得对 CPU 的控制。如下文所述，在某些无磁盘（disk-less）、无介质（media-less）的工作站上，位于网络接口卡上的适配器固件为引导操作系统的指针重定路径，从附接的网络下载操作系统。

BIOS 进一步包含一组例程或中断处理程序用于与计算机及其外围部件接口。BIOS 中断处理程序是通过使用硬件或软件中断被访问的。这些中断处理程序的地址被存储在一个中断向量表中。如上所述，这个向量表可被修改为指向定制的中断处理程序。P.Norton 在《The Peter Norton PC Programmer's Bible》（Microsoft 出版社，1993）中概括地说明了 BIOS。

尽管 BIOS 接口例程为 MS-DOS 操作系统所使用，现代操作系统如微软公司（microsoft）的 Windows-95 却不广泛使用 BIOS 接口例程。

Windows-95 和其它现代操作系统利用特定类型和模型的外围硬件部件专用的设备驱动程序来与这类外围硬件部件通信。设备驱动程序提供一个更通用的软件可通过其与外围部件交互作用的统一接口。这些设备驱动程序可以取代现有的 BIOS 中断处理程序，或提供否则就没有的另外的功能。应用软件于是就免于不得不与这种硬件设备的具体细节交互。

许多操作系统，包括 MS-DOS 2.0 以后的 MS-DOS 版本以及直到 Windows 3.11 的 Windows 版本，都包括有在操作系统被引导时从磁盘加载可安装的设备驱动程序的能力。在 MS-DOS 操作系统中，用户通过在 CONFIG.SYS 文件中加入 DEVICE=device_file 命令就可以加载可安装的设备驱动程序。MS-DOS 然后读取每个设备驱动程序文件，将设备驱动程序装入内存。Windows-95 能够用 PCI（外围部件互连）和 BIOS 的即插即用功能来检测外围硬件部件，自动地为所安装的外围硬件部件加载适当的驱动程序。

尽管可安装设备驱动程序的使用为管理外围硬件部件—诸如网络接口卡（NIC）—提供了高度的灵活性，迄今为止，都是依赖在故障计算机上执行操作系统来加载软件驱动程序和提供必需的支持功能。如果操作系统不引导，或者有必要在重新加载操作系统之前进行故障后诊断，就不会为 NIC 加载软件驱动程序，于是管理员将不能使用通过 NIC 工作的软件来远程访问系统。因此，需要一种使用在操作系统被引导之前可以得到的、不依赖操作系统的可安装设备驱动程序的方法和系统。本文中所用的操作系统一词，意思是控制用户级程序的执行并对这种用户级程序提供诸如资源分配、调度、I/O 控制和数据管理等服务的系统级软件。这种操作系统的典型例子是 MS-DOS、Windows-95、Windows-NT（都可从 Microsoft 公司得到）、MacOS（从 Apple Computer 公司得到），以及可从包括 Sun Microsystems 公司的许多销售商得到的各种版本的 Unix。现代操作系统诸如 Windows-NT，通常在操作系统的核心有一个有保护模式的内核或基系统。

计算机系统的远程管理的一个关键问题是，事实上有可从许多销售商得到的数百种不同的网络接口卡类型，每一种都可能有不同的编程方式，都可能使用独有的设备驱动程序。为这些卡类型的每一种开发新的设备驱动程序会花费巨大，而且导致不可靠性。因此本发明的

一个目的是利用网络增强的 BIOS 来使用为现有操作系统开发的标准 NIC 设备驱动程序，这样就不要求每一种可得到的网络接口卡类型都有定制的设备驱动程序。

5 这个目的可以通过使用由某些操作系统销售商定义的标准接口来实现。为了支持几乎所有各种网络卡类型，操作系统销售商已经定义了要由网络接口卡设备驱动程序使用的标准接口。这就使操作系统能支持任何提供符合标准接口的驱动程序软件的 NIC。Novell 公司 (Novell) 已经定义了一个这样的标准，即开放式数据链接接口 (ODI - Open Datalink Interface)。按 ODI 标准编写的驱动程序可以被
10 Novell 提供的 NetWare™ 使用。微软公司已经定义了第二个标准 - 网络驱动程序接口规范 (NDIS)。按 NDIS 编写的驱动程序可以被微软公司的操作系统 (例如 Windows NT™) 使用。此外还有其它的标准可用于 UNIX™ 操作系统的各版本。

除了以上列举的目的，本发明的一个目的是提供一种无需使用计
15 算机的操作系统能与连接到网络的计算机通信的方法和系统。

本发明的另一个目的是提供一种在计算机的操作系统的引导之前能与连接到网络的计算机通信的方法和系统。本发明的再另一个目的是提供计算机上必要的更多软件和可在远离第一个配备网络增强型 BIOS 的计算机的第二个计算机上执行的软件，第二个计算机本身不
20 必配备这种网络增强型 BIOS 就能在这两个计算机之间传输命令和信息。

本发明的目标是一种在引导计算机的操作系统之前或操作系统发生故障之后通过网络与计算机通信的方法和系统。在第一个计算机配备的网络增强型 BIOS 中实现一个连网应用程序接口 (API)。这个计
25 算机最好配有 NIC 卡和 NIC 设备驱动程序文件。在 NIC 设备驱动程序中通常被解析到由操作系统的单元提供的服务的外部引用转而被解析到引用由网络增强型 BIOS 的 API 提供的服务。通过网络连接到第一个计算机的第二个计算机可以在操作系统加载之前或者操作系统故障之后传输命令、状态和数据。第一个计算机可以进一步配备在发生
30 发生 POST 故障或操作系统崩溃时向第二个计算机报警的手段。

要更彻底地理解本发明，可参看下文结合以下各附图所作的详细说明：

图 1 是示例性系统结构的功能框图，该结构包括服务器计算机、台式计算机、连网的管理工作站和实施本发明的网络；

图 2 是实施本发明的包括计算机系统部件的示例性计算机系统的功能框图；

5 图 3A 是计算机的常规 BIOS 部分的功能框图，图中展示了按本发明创建的 BIOS 的一部分的主要功能部件；

图 3B 是计算机的网络增强型 BIOS 部分的功能框图，图中展示了按本发明的主要功能部件；

10 图 4 是表示在按照本发明的最佳实施例为远程访问配置的计算机上含有的软件的流程图；

图 5 是按照本发明的网络接口部件的各层的功能框图；

图 6 是按照本发明的最佳实施例由网络增强型 BIOS 实现的网络协议堆栈的功能框图；

图 7 是按照本发明实现的 RPC 设备的功能模型；

15 图 8 是按照本发明处理 BIOS 模式之间的转换的机制的功能框图；

图 9 是按照本发明的监视计时器机构的功能框图。

现在将连续参考各图来说明本发明的最佳实施例。

图 1 表示系统体系结构最佳实施例的总体图。服务器计算机 100 通过网络 300 与管理工作站 200 实际相连。网络 300 可以是包括
20 Novell 公司的 NetWare™ 在内的许多市场可购得的局域或宽域网中的任何一个。本文中所用的术语“网络”或“计算机网络”一般将定义为一组两个或更多相连的计算机。如果计算机能交换数据，则称它们是“相连”的。因此例如，连接技术、网络拓扑和网络协议可能是不同的，例如包括但不限于—除上述 LAN 和 WAN 网络以外的—通过串行
25 或并行接口和电缆直接相连的两个计算机、通过公共交换电话网（诸如通过“简易老式电话服务”）上的调制解调器连接而直接连接的两个计算机、通过公共电话网上的 ISDN 连接而连接的两个计算机、在互连网上相连的两个计算机。台式计算机 400 也可以与网络 300 相连。所示的服务器计算机和台式计算机是示例性的，可以是多台这类
30 计算机中的一台。

现在参看图 2，所示的是可在其上实施本发明的计算机 400。图 2 所示的本文所描述的计算机 400 是示例性的，可以在不偏离本发明的

情况下作修改。此外，尽管图 1 中所示的计算机 400 是台式计算机，应当明白本文中
所用计算机一词应包括—但不限于—台式计算机（图 1 中所示的 400）和服务
器计算机（图 1 中所示的 100）或内置系统（未予表示），诸如自动柜员机、
销售点终端或销售机。

5 系统总线 105 构成了计算机 400 的部件的主干。与系统总线 105 连接的是中央处理单元（CPU）110 和系统随机存取存储器（RAM）120。CPU 最好以至少两种方式操作，即实模式和保护模式—这在本领域是公知的。非易失性存储器 125—最好是快闪 ROM 的结构，与系统总线 105 相连。非易失性存储器 125 在计算机 400 关机或复位时不改变状态。用户输入/输出可通过可视显示器 130、视频控制器 135、键盘 140 和键盘控制器 145 来进行。CPU 110 和 RAM120 以及视频控制器 135 可任选地通过单独的局部总线（未予示出）连接。海量存储设备可包括软盘驱动器 150 和关联的软盘控制器 155，和/或硬盘驱动器 160 和关联的硬盘控制器 161。在下文说明的实现监视计时器机构的本发明的一个实施例中，计算机 400 也包括系统管理 RAM（SMRAM）121、外部计时器 122 和芯片集外部逻辑 123。计时器 122 最好是在芯片集逻辑 123 内部实现。如上所述，图 2 的说明是示例性的，可以按本领域的熟练人员显而易见的许多方式修改而不偏离本发明。例如，本发明可以在没有软盘驱动器 150 或硬盘驱动器 160 的无磁盘的工作站上实现。

计算机 400 也最好包括网络接口卡（NIC）170—也称作网络适配器。NIC 170 最好通过计算机内的一组系统线路插槽 165 连接到系统总线 105。插槽 165 最好符合某工业标准，诸如 Intel 的 PCI 标准。或者，NIC 170 可以不用系统线路插槽而直接与系统总线 105 连接。

25 或者，计算机 400 可以用调制解调器代替 NIC 170 或除 NIC 170 外还使用调制解调器。这种调制解调器可以是内置式调制解调器 190—它可以连接到系统总线 105 或该组系统线路插槽 165 之一、和/或外置式调制解调器 180—它一般连接到与系统总线 105 相连的串行接口 175 上。可选地，计算机 400 可以通过一个连接串行接口 175 或并行接口（未予示出）的电缆连接直接与工作站 200 连网。

现在参看图 3A 和 3B，所示的是计算机的常规 BIOS 区域 500 和按照本发明创建的网络增强型 BIOS 区域 600 的部件模块。常规 BIOS

区域的组成是早期 POST 510、其余 POST 520、显示服务 530—诸如在 BIOS 内部用于显示错误和状态信息的光标位置和写字符、POST 时间服务 540—诸如硬件测试例程、运行时服务 550—诸如中断 10H 视频输出服务和中断 13H 磁盘存取服务、以及启动程序 (enabler) 代码 560。启动程序代码 560 使常规 BIOS 500 能检测和加载网络增强型 BIOS 600，并将控制交给网络增强型 BIOS 600 的初始化入口点。它进一步含有允许在常规 BIOS 500 与网络增强型 BIOS 600 之间有合作性多任务的例程。早期 POST 510 与其余 POST 520 之间的划分是一种实用划分，POST 被划分为大量的在 POST 调度程序例程 505 控制下执行的例程。早期 POST 510 最好是为将计算机充分地配置成使网络增强型 BIOS 能初始化而要求的那组 POST 例程。如下文所述的那样，早期 POST 510 最好包括的操作有：编程内存控制器，确定内存的大小，测试内存，充分地初始化芯片集以访问如果存在的 PCI 总线，初始化和测试中断控制器，为网络适配器配置中断，初始化如果存在的 VGA 控制器。早期 POST 510 也可以任选地编程其它芯片集寄存器。早期 POST 例程 510 最好存储在非易失性存储器 125 中。

网络增强型 600 用初始化区域 610、集成了内核调度程序 621 的 32 位保护模式的内核 620 和加载器 630 来加强常规 BIOS 区域 500。网络增强型 BIOS 600 进一步用利用该内核所提供服务的协议堆栈 640 来加强常规 BIOS 区域 500。由于现代操作系统一般包括它们自己的一个保护模式的内核，网络增强型 BIOS 600 一般在操作系统的引导后就终止。然而本发明的变例允许至少有一部分网络增强型 BIOS 600 留在内存。这个驻留部分用于与下文描述的监视器检测机构接口。

现在参看图 4，系统启动 410 后，CPU 控制被传送 411 到早期 POST 510 例程。早期 POST 510 执行诸如编程内存控制器、中断控制器、系统总线控制器以及其它芯片集寄存器等系统操作所必需的基本任务。这个操作与不是网络增强型的 BIOS 的操作相同。

控制然后转移 412 给网络增强型 BIOS 的安装检测例程，它是图 3 的启动程序 560 的一部分。安装检测例程在非易失性存储器 125 中顺序扫描，寻找被称为标记的、指示网络增强型 BIOS 600 代码的开始的一个数据序列。如果安装检测例程确定 439 不存在这种标记，则其

余 POST 例程 520 执行 413, 控制被传送 414 给由操作系统提供的引导例程。如上所述, 引导例程装载 432 操作系统并向它传送 415CPU 控制。

5 如果找到指示网络增强型 BIOS 600 代码的开始的标记, 则重定位例程 416 就将网络增强型 BIOS 600 从非易失性存储器 125 复制到 RAM 120, CPU 控制被传送 417 到位于相对标记的开始的一个已知偏移地址的网络增强型 BIOS 600 的初始化例程 610。由于常规 BIOS 510 通常是以实模式操作的, 网络增强型 BIOS 初始化例程 610 将 CPU 转换成保护模式并初始化 32 位内核 620 以及协议堆栈 640 的各个部件。

10 如图 4 中所示, 网络增强型 BIOS 内核 620 最好是多线程的, 就是说, 它通过将 CPU 控制按哪一个任务具有最高优先权而从一个任务切换到另一个任务来支持看起来同时执行多个任务。如果任务正在等待某个外部事件或等待它们所依赖的另一个任务完成执行, 它们的执行就会受阻塞。为帮助任务之间的合作, 内核提供信号、信号量和互相排斥功能的功能, 对此本领域的熟练人员是熟悉的。CPU 控制在不同任务之间的切换是由内核调度程序 621 执行的。

20 网络增强型 BIOS 初始化例程 610 此时启动许多线程来支持网络协议堆栈 640, 以及不同的与管理工作站 200 应用程序通信的特征。这些线程最好是 32 位保护模式的代码, 将在下文进一步解释这些特征时讨论。

现在将描述常规 BIOS 500 与在本发明的第一个实施例中采用的网络增强型 BIOS 600 交互操作的方法。该交互操作的方法的细节可以在不影响本发明原理的情况下进行修改。

25 参看图 8, 图中进一步详细地展示了从常规 BIOS 代码到网络增强型 BIOS 600 的初始转换, 它如随后发生的交互操作切换的重复过程一样。BIOS 保存其状态 805, 即将所有 CPU 寄存器和标志存储在内存数据区。网络增强型 BIOS 600 代码和数据然后被解压并重定位 810, CPU 控制被转移到这个代码的入口点 815。内核及网络协议堆栈被初始化 820。处理器被返回到实模式 825, 调用一个常规 BIOS 服务程序
30 将与 BIOS 的内核模式转换入口点的地址登记 830。这个例程返回 BIOS 模式转换入口点的地址 835。

然后进入一个重复状态, 在这个状态下, 常规 BIOS 和 32 位内核

线程交替运行。内核保存其状态，即将所有 CPU 寄存器和标志存储在内存数据区。然后执行向 BIOS 模式转换入口点的转移 845。以前保存的 BIOS 状态被恢复 850，常规 BIOS 代码执行。CPU 控制被传送 855 给常规 BIOS 区域 500 的 POST 调度程序例程 505，后者将执行当前的 BIOS POST 例程，一直到它要么完成，要么由于等待诸如中断的外部事件、或编程的延时结束而挂起。某些 POST 任务诸如内存检测可能执行的时间较长，在这些任务内，常规 BIOS 将定期地将控制交给 32 位内核 620。此时 BIOS 再次保存其状态 860，并执行到内核模式转换入口点的转移 865。内核恢复其保存的状态 870，重新进入保护模式 875，内核调度程序 621 然后将调度所有没有被阻塞的线程 880。当内核调度程序 621 确定没有准备好要执行的线程时，即系统“空闲”时，内核重新进入实模式 885，这个过程从内核保存其状态 840 这一点重复地进行。

内核 620 最好这样来实现中断处理，使得如果 BIOS POST 例程正等待中断并因此已经交出控制，当在 32 位内核任务期间发生中断时，该中断被反映到 BIOS 任务。反过来也是一样，为 32 位内核线程设计的中断，当在 BIOS 任务执行期间发生时，必须被反映到该 32 位内核任务。中断反映的管理对于了解微软视窗虚拟设备驱动程序 (VxD) 的人来说是熟悉的。处理器模式转换的性能在 Intel 公司发表的微处理器手册中有记载。根据本公开的技术，对其它驱动程序类型和微处理器体系结构的类似解决方案，对于本领域的熟练人员来说是显而易见的。

常规 BIOS 和 32 位网络增强型 BIOS 任务交替执行的作用是，网络增强型 BIOS 线程与当前的常规 BIOS POST 任务看起来是同时操作的。返回图 4 可以看到，在初始化 417 之后有两个执行路径，在会话键 (下文讨论) 已经交换 422 之后有多个执行路径。概念上讲这些任务和线程是同时操作的。图 4 中的虚线表示线程 432 到 427 的同时执行。

本文用术语“任务”来描述本发明的最佳实施例，该术语概括地表示来自在由 POST 调度程序顺序执行的常规 BIOS 的实模式例程的序列。某个时刻这些任务只有一个是活动的。本文所用的术语“线程”，概括地表示由内核调度程序 621 正在调度的 32 位保护模式的代码。

这些线程概念上与其它线程以及单一的常规 BIOS POST 任务是同时运行的。然而对于本领域的熟练人员来说，显然本发明并不受限于任务与线程之间的任何区分。

参看图 4, 常规 BIOS 区域 500 继续执行 418 完其余 POST 例程 520。
5 网络增强型 BIOS 600 现在“同时地”开始使用网络通信。下文要给出网络协议堆栈操作的细节，这一节给出的是该事务发生的概述。网络增强型 BIOS 600 先发出 419 一个报警分组到网络 300，准备打开一个数据连接。网络增强型 BIOS 600 以预定时间等待来自管理工作站 200 的应答。这个报警分组和任何可以随后发出的格式最好是一个
10 SNMP（简单网络管理协议）分组。SNMP 是 TCP/IP（传输控制协议/因特网协议）组合内部的一种标准/协议，通常用于管理和监控 TCP/IP 网络上节点。SNMP 和 TCP/IP 在本领域是众所周知的，例如在 J. Martin 所著的《TCP/IP Networking》（PTR Prentice Hall, 1994）就有描述。即使没有管理工作站应答，这也允许在连接在网络
15 上的计算机上运行的诸如可从 Hewlett-Packard 购得的 HP OpenView™ 等第三方管理软件来记录及显示这些报警。其它协议的使用对于本技术中的普通技术人员是显而易见的。

如果在一个可配置的时间期间之后（它最好是 5 秒的数量级）仍未收到来自管理工作站 200 的应答，可将网络增强型 BIOS 600 配置
20 成或者试图对不同的管理工作站进行连接，或者可以终止该网络增强型 BIOS 600 的操作，在这一情况中，传统的 BIOS 部分 500 将以正常方式完成 POST。如果没有 POST 错误，传统 BIOS 部分 500 将发布 431 引导报警分组到网络 300 上并将 CPU 控制传送 414 给 BIOS 引导例程，从而给引导扇区代码 432，从而给操作系统 415。如果 POST 520 例程
25 的其余部分检测到 438 错误，它将判定 437 该错误是否是关键性的。如果 POST 520 例程的其余部分检测到非关键性错误，则传统 BIOS 部分 500 将发布 434 非关键性错误报警到网络 300 上并继续处理。POST 520 例程的其余部分如果检测到 437 关键性错误，则它将发布
30 435 致命错误报警到网络 300 上并进入 436 诊断模式。

如果接收到来自管理工作站 200 的应答，则最好通过诸如计算基于消息内容的散列函数及只应是授权的管理工作站 200 应用及具有网络增强型 BIOS 600 的计算机 400 知道的秘密密钥来验证 421 它。

将其与作为工作站的应答的一部分发送的消息验证代码进行比较。只有授权的管理工作站才能生成包含正确的验证代码的应答。

5 工作站 200 与网络增强型 BIOS 600 之间的进一步通信最好加以验证，诸如利用秘密密钥来交换 422 基于随机数的会话密钥。可利用公知的密钥交换机制之一，诸如在 Steiner 的“应用密码学”中所描述的 Diffie-Hellman。也可参看 Hellman 等人的美国专利 4,200,770 号。对于其余会话来说，要求所有将来的网络响应都使用交换钥来计算消息验证代码。

10 一旦交换了会话钥，就创建其它的线程来提供待由管理工作站支持的功能所要求的另外的服务。为了提供远程控制台设备，启动键盘重定向线程 423，它接受发自管理工作站 200 应用的扫描代码，用通常为诊断目的设计的机制将扫描代码放入键盘控制器 145 的输出缓冲区。这些扫描代码然后被计算机 400 的标准诊断服务例程读取，其效果等同于在键盘 140 上按下一个键。键监控例程 430 检查在键盘上接收或击打的所有按键。某些键被解释为是要执行特殊动作—诸如进入特殊诊断模式或执行特定类型的再引导操作—的指令。

20 屏幕重定向线程 424 通过截获 BIOS 屏幕显示中断 10H 以及 BIOS 显示服务输出而捕获所有对屏幕的输出。输出被缓存，并在管理工作站应用请求传送未完成屏幕数据时定期发送给它。屏幕重定向线程 424 和键盘重定向线程 423 在下文结合应用协议层作进一步说明。

25 磁盘重定向线程 425 提供支持对软盘驱动器访问的重定向的选择。如果是这样配置的，BIOS 软件磁盘服务中断（Intel 80×86 上的中断 13H）就被截取，所有对软盘驱动器（通常为 A: 驱动器）的引用都被缓存，并在管理工作站 200 应用请求要传送未完成磁盘操作时定期发送给它。为服务于这些引用，管理工作站引用访问其自己的软盘、其硬盘上的文件、或位于网络上其它地方的另一个计算机上的驱动器上的文件。BIOS 磁盘重定向线程 425 的操作将挂起，直到重定向的磁盘操作发生，且数据或结果已经通过网络 300 返回。作为中断 13H 操作的结果，所传输的数据被返回到调用应用或者操作系统函数。

30 上述磁盘重定向机制可用来向配备网络增强型 BIOS 的计算机提供从磁盘或另一个计算机上驻留的文件引导操作系统的能力。这在整

个磁盘子系统都发生故障的情况下诊断计算机上故障特别有用，并且还可用于建立无磁盘的工作站计算机。

如上文介绍本发明的背景时所述，这个功能以前在 PC 兼容计算机上就有了，方法是安装一个具有特定网络卡专用的“引导 ROM”的网络卡，这需要额外的花费。按照本发明，这个从远程设备引导和访问远程设备的功能是网络增强型 BIOS 600 的一个通用性功能，能用在各种有诸如 ODI 或 BDIS 驱动程序等标准驱动程序可用的各种网络卡上。

加载器服务线程 426 向管理工作站 200 应用提供一个到模块加载器 630 的接口。管理工作站 200 能向含有要装入 RAM 并作为另外的瞬态网络增强型 BIOS 线程执行的代码与数据的模块加载器 630 发送消息。

异步命令线程 427 向管理工作站 200 提供一个接口，通过它能向网络增强型 BIOS 600 发出异步命令，例如请求传输 BIOS 数据、进入特别诊断模式或执行特定的再引导操作。

以上描述的所有线程在网络增强型 BIOS 600 处于活动的整个期间连续运行，通常一直到 POST 完成。如上所述，如果没有管理工作站 200 的介入，并且没有 POST 错误，就发送一个指示 POST 成功完成的报警 431。

网络增强型 BIOS 600 可以进一步安装其自己的处理计时器滴答中断（80x86 微处理器上的中断 08H）的中断处理程序。计时器滴答中断向量指向图 8 中的实现步骤 860 的过程，该过程迫使当前实模式保存其状态并转移到网络增强型 BIOS 600。这使网络增强型 BIOS 600 能在诸如 MS-DOS 的某些实模式操作系统加载和运行期间以一定的间歇继续获得 CPU 控制。注意，BIOS 屏幕显示中断 10H 此前可能已经被重定向到一个作为已经说明过的远程控制台设备的一部分的处理程序，这个机构依然在位。CPU 控制然后传送到引导程序 432。

如果要加载的操作系统是保护模式的操作系统，诸如 NetWare 或 Windows-NT，则它将把处理器转换到保护模式，置换包括上述的计时器滴答中断和屏幕显示中断的中断向量，取得对 CPU110 的控制，除了其崩溃恢复诸功能外，网络增强型 BIOS 将停止工作。

如果加载的是没有任何保护模式内存管理器的实模式操作系统诸

如 MS-DOS, 则网络增强型 BIOS 代码将留在扩展内存中。当 MS-DOS 重新编程中断向量时, 它将链接现有的向量, 使得 MS-DOS 中断处理例程将首先执行, 然后执行该向量在 MS-DOS 引导程序启动时所指向的例程。这样的结果是, 包括远程控制台设备在内的网络增强型 BIOS 600 的所有功能, 在这种操作系统运行时继续是可用的。

对本领域的熟练人员来说, 显然, 根据本公开中的介绍, 可以在 POST 时或 DOS 环境中添加更多的可以在内核调度程序控制下运行的线程。

网络协议层

10 现在参看图 5, 要说明的是一个提供 BIOS 内部网络接口的最佳方法。

网络支持一般被划分成几个软件层。例如, 国际标准组织为网络支持定义了一个七层模型。该划分中的每一层都可以使用由它“下面”的一层所提供的服务, 并向“上面”的一层提供服务。

15 网络驱动程序层 724 是底部软件层。它下面的层是连网软件本身—网络控制器接口 725。网络驱动程序层 724 向上面的连网协议层 722 提供服务。

20 连网协议层 722 本身在逻辑上分为若干层, 其中较低层协议用于使更高层协议能操作。例如, 一个投递分组数据但不保障投递或投递次序的无连接的数据报协议(例如不可靠的数据报协议(UDP)—图 6 的 732)被支持有保障的顺序数据投递的更高层协议(例如传输控制协议(TCP))使用。本发明的最佳实施例只实现无连接数据报协议来节省非易失性 RAM 的空间。UDP 732 传送 IP 数据报或分组到 IP 层 733 (图 6)。IP 层 733 的功能是使网络 300 的拓扑和物理特征透明化。

25 模型的最顶一层是应用层 721, 它用协议层 722 所提供的服务来在网络上传输数据。

网络驱动程序层

30 网络驱动程序层 724 将网络卡硬件 725 与连网协议层 722 相连。网络驱动程序层 724 最好按照标准的接口规范来实现。如上所述, 其中的两种可能规范是 ODI 和 NDIS。本文说明的实现详细描述了按照 ODI 标准编写的驱动程序的使用。对于本领域的熟练人员来说, 实现一个支持使用按其它标准—诸如 NDIS 或 UNIX 标准—开发的驱动程

序、甚至支持多个标准网络增强型 BIOS 是显而易见的。

现在参看图 6，将说明网络协议堆栈的最佳实施例。ODI 规范定义了一个将网络设备驱动程序 724 与诸上层隔离的接口。链路支持层 (LSL) 734 向网络堆栈和网络驱动程序这两个上层提供登记和缓冲器分配服务。网络设备驱动程序的诸部件被统称为多级链路接口驱动程序 (MLID)。MLID 738 本身可以划分成三个部件：媒体支持模块 (MSM) 735、拓扑特定模块 (TSM) 736 和硬件特定模块 (HSM) 37。MSM 例程 735 是所有驱动程序公用的，支持网络驱动程序层 724 与网络增强型 BIOS 的接口。TSM 例程 736 对用于诸如以太网或令牌环等特定连网技术的驱动程序是公用的。HSM 例程 737 是每个特定 NIC170 特有的。HSM 例程 737 必须处理 NIC 初始化、开机、关闭、复位、分组接收与传输、超时。

为了便于网络卡驱动程序的开发，许多对许多 TSM 和 MSM 例程来说公用的代码都被第三方开发商标识出来放入能被驱动程序使用的常用例程集中。这些例程可以移植到 BIOS 中，没有操作系统的支持也能执行。这些 TSM 和 MSM 例程又可被 HSM 驱动程序 737 所依赖。按照定义，符合 ODI 的 HSM 设备驱动程序必须只使用 ODI 规定的服务。因此，支持那些 TSM 和 MSM 服务的网络增强型 BIOS 600，无需操作系统的帮助就将能支持符合 ODI 的 HSM 设备驱动程序。

如上所述，MSM 和 TSM 例程在网络增强型 BIOS 600 中的实现最好通过添加一个小型多任务内核来进行。

网络增强型 BIOS 600 最好实现 LSL、MSM 和 TSM 例程。然而，可以省略这些例程中的许多功能，以节省它们最好在其中存储的非易失性 RAM 125。这样，网络增强型 BIOS 600 向连网接口卡 170 提供一种与操作系统所提供的类似的环境，允许符合现有标准的设备驱动程序无需修改就能实现本文目的的功能。最优的最小例程集的确定方法是，检查可从不同制造商得到许多不同的 ODI 驱动程序，确立哪些例程是要由网络增强型 BIOS 支持的应用层服务所必需的。本发明人目前所理解的用于 NE2000、Intel EtherExpress 和 3Com 90x 的最佳引入功能集，如下面的表 1 所列：

表 1

EtherTSMFastProcessGetRCB	MSMParseCustomKeywords
EtherTSMFastRcvComplete	MSMParseDriverParameters
EtherTSMFastSendComplete	MSMPrintString
EtherTSMGetASMHSMIFLevel	MSMPrintStringWarning
EtherTSMGetNextSend	MSMRdConfigSpace16
EtherTSMGetRCB	MSMRdConfigSpace32
EtherTSMRegisterHSM	MSMRdConfigSpace8
EtherTSMSendComplete	MSMReadPhysicalMemory
EtherTSMUpdateMulticast	MSMRegisterHardwareOptions
GetCurrentTime	MSMRegisterMLID
MSMAlertFatal	MSMReturnDriverResources
MSMAlertWarning	MSMReturnRcvECB
MSMAllocPages	MSMScanBusInfo
MSMAllocateRCB	MSMScheduleIntTimeCallBack
MSMDriverRemove	MSMSearchAdapter
MSMFreePages	MSMSetHardwareInterrupt
MSMGetMicroTimer	MSMWrtConfigSpace16
MSMGetPhysical	MSMYieldWidthDelay
	IOConfigurationList

协议层

为便于应用开发，网络增强型 BIOS 600 最好也实现一些不同的协议层。最好实现两个系列的协议：基于 Netware™ 的网络公用的因特网分组交换/有序分组交换（IPX/SPX）—连同其所要求的较低层协议；传输控制协议/因特网协议（TCP/IP）协议系列—连同其支持协议诸如因特网控制消息协议（ICMP）。然而，在非易失性 RAM 非常宝贵的情况下，该实现在功能上可以部分或全部省略这一点。例如，网络增强型 BIOS 600 可以实现 IPX 而不实现保障分组投递且在 IPX 的顶上实现的 SPX。

10 RPC 层

为进一步便于应用开发，最好实现一个提供远程过程调用（RPC）731（图 6）的功能的协议层。这是例如由 Nelson B. J.、Birrell A. D. 所描述（见《Implementing Remote Procedure Calls, ACM

Transactions on Computer Systems》2(1), 1984年2月)的RPC的功能的一个子集。如图7所示, RPC服务器751配备在网络增强型BIOS内, RPC客户机752配备在管理工作站200应用内。这就使管理工作站200上的应用程序能包括一个简单功能调用753, 其参数是用较低层网络协议在网络300上传送给带有网络增强型BIOS 600的机器400的。RPC功能然后由该网络增强型计算机400上的一个服务例程755执行, 结果被返回756到管理工作站。客户机调用过程753和服务器被调用的过程可以分别与一个调用占位程序或被调用占位程序(未予示出)接口。这为应用程序能与网络增强型计算机400交互作用提供一种简易的方法。网络增强型计算机400最好用诸如上文已说明过的消息验证代码协议来验证RPC请求。

RPC机制也为OEMS实现增强的服务—诸如从服务器管理硬件访问数据或在制造测试时装载诊断代码—提供了手段。

应用层

应用层721含有管理员可远程访问的各种应用服务。

这些应用最好包括屏幕和键盘向网络300的重定向, 允许应用的控制经过网络, 而不是使用本地键盘140和视频显示器130。这将提供远程控制台设备许多特征, 诸如允许远程控制台设备观察由POST报告的消息和任何错误以及用BIOS建立屏幕或其它BIOS配置方法来允许查看和改变由BIOS控制的参数。

屏幕和键盘通过网络300的重定向可以用上述的线程423和424(图4)及RPC机制来取得。键盘重定向的取得方法是, 捕获工作站200的按键, 使RPC调用将对应于按键的扫描代码从工作站200传输到计算机400。网络增强型BIOS内的服务线程423然后模拟从本地键盘接收的扫描代码的效果。实现这种模拟的其中一个方法是, 向键盘控制器电路145发送一个特殊命令, 将所传输的扫描代码插入键盘控制器电路的输出缓冲器。

屏幕重定向更复杂, 因为它要求从网络增强型BIOS 600向工作站200传输数据, 而所述RPC机制一般不允许这种传输从计算机400启动。因此在工作站200应用程序中要调用RPC功能, 在网络300上传输一个请求, 该请求然后启动计算机400上BIOS的网络接口内安装的一个服务例程, 阻塞各屏幕重定向线程的执行, 直到有屏幕数据准

备好向工作站 200 发送。如上所述，服务最好是多线程的，以便在等待屏幕数据期间让其它操作继续。服务例程通过挂钩 BIOS 屏幕显示服务中断就能捕获屏幕数据，或者它可以直接链接到向屏幕输出字符的 BIOS 显示服务代码。当屏幕数据可用时，它被放置在内存的缓冲区，这导致 RPC 服务器的服务线程被解除阻塞。发送一个含有屏幕数据的网络分组。这导致 RPC 客户机线程解除阻塞，屏幕数据作为原始功能调用的结果被返回给管理工作站。

其它采用最佳分层体系结构、特别是 RPC 功能的可能应用包括诊断例程，这些例程允许通过发出测试命令和浏览使用网络的工作站 200 上的结果来测试网络增强型计算机 400。

能采用最佳分层体系结构的其它可能应用对于本领域的熟练人员来说是显而易见的。

二进制兼容性

网络适配器制造商通常与每个网络适配器一起提供含有操作系统特定的设备驱动程序的文件。为了使用为这种操作系统开发的网络驱动程序，网络增强型 BIOS 600 必须使用驱动程序文件内含有的二进制映象。按照 ODI 标准，驱动程序映象的格式是 NetWare Loadable Module (NLM) 文件格式。网络适配器制造商提供的支持 ODI 标准的驱动程序实现 ODI 规范的 HSM 部分。加载进程必须转换这个格式并将驱动程序装入非易失性存储器。在加载进程中，设备驱动程序引用外部例程和其需要被处理的入口点。NLM 格式含有关于为支持其含有的代码的操作所需要的外部例程的信息。信息的形式是一个表，内容是外部例程必须由此调用的地址，和可以解释为是外部例程的名称的对应引用。在加载进程期间，所需要的例程位于内存中，需要引用这些外部例程的 NLM 中的位置被改变。这个过程对本领域的熟练人员来说是熟悉的，他们一般称之为“链接”。

在本发明的一个实施例中，一个基于独立操作系统的实用程序读取 NIC 制造商提供的驱动程序文件，解析对驱动程序中外部符号的引用，输出一个然后被装入非易失性存储器的二进制文件。在本发明的另一个实施例中，驱动程序文件将被 BIOS 代码直接装入非易失性存储器。网络增强型 BIOS 600 在系统启动时解析对驱动程序中外部符号的引用。如上所述，由制造商所提供设备驱动程序引用的外部例程

是在网络增强型 BIOS 内部实现的，这就消除了访问操作系统的需要。

除了实际的驱动程序代码和数据之外，驱动程序二进制文件还含有几个表，包括：

- 5 · 一个描述各种其它表的位置和性质的文件首部；
- 驱动程序代码段的映象；
- 驱动程序数据段的映象；
- 应当由其它模块（导入表）提供的符号的名称和使用位置；
- 驱动程序提供的能被其它模块使用的符号的名称和位置；
- 10 · 一个能根据驱动程序被装载到的最终地址固定驱动程序代码和数据映象的表，这个表通常被称为“组织”表。

驱动程序加载代码（要么是网络增强型 BIOS 600 的一部分，要么是基于独立的操作系统的实用程序中的）用这些表中的信息，来解析对其它模块的引用，根据最终的加载地址来定位驱动程序映象。

15 用调制解调器替代 NIC 的操作

如上所述，在不可能有使用 NIC 的网络连接的情况下，可能需要用调制解调器连接作为替代。实现的方法最好是，创建一个软件模块，它实现 ODI 规范的 MLID 部分，但被改造为适合通过计算机 175 的串行接口——因此就是通过一个连接远程计算机的外置式调制解调器 180 来发送和接收数据。（图 2）或者也可以使用内置式调制解调器。这个软件模块作为 HSM 的替代，按上述方法在非易失性存储器中链接和安装。

在另一实现中，支持调制解调器的软件模块最好设计成与协议堆栈的 UDP 层一样的 API。在这个情形中，UDP 以及协议堆栈的所有较低层都可以从计算机的非易失性存储器中省略。

25 操作系统故障检测和处理

操作系统 901（图 9）的故障可以通过许多已知技术中的任何技术或通过下文描述的新的监视机制来检测。

许多操作系统故障导致一个有控制的故障模式，在这种模式中，操作系统本身检测故障并执行错误例程。可以检查操作系统代码或文档来确定这种错误例程的执行的途径。通过使用具有系统级特权的设备驱动程序，就有可能修改错误例程，迫使它将控制交给为此目的保

留在内存中的网络增强型 BIOS 600 的一部分。网络增强型 BIOS 最好也提供应用程序接口 (API) 或供操作系统在进入受控制的故障代码时寻址的简单错误例程入口点。这在操作系统采用能用于指示将其错误处理程序例程定向到外部代码的 API 的情况下是有用的。

5 其它用于检测操作系统故障的技术包括使用监视计时器装置。在 Burckhartt 专利中描述了一例监视计时器装置，它与在操作系统控制下运行的一个程序联合操作。操作系统在正常操作期间定期地再触发计时器。如果 Burckhartt 专利中描述的计时器的时间段到期，没有程序重新触发计时器，就假设操作系统发生了故障。计时器的输出直接或间接地连接处理器的复位引脚，使系统复位。这就阻止了对关于故障的原因的有用信息的收集。

现在参看图 9，本发明的监视计时器机制（下文说明）克服了这些缺陷，最好用于网络增强型 BIOS，因为它允许采用网络增强型 BIOS 的数据传输方法。对故障诊断有用的信息可以被捕获并传送到远离故障机器的系统管理器。这与现有技术不同，因为事实上监视计时器机制不立即将机器复位，反而让某些基本数据收集操作首先发生。采用的是微处理器的系统管理模式 (SMM)。数据收集例程 920 被网络增强型 BIOS 安装到系统管理 RAM (SMRAM) 121 (图 2) 中。这个数据收集 920 例程可以在上述许多步骤中的任何步骤期间安装，例如在图 8 的内核初始化步骤 820 期间安装。SMRAM 121 只有在系统管理中断 (SMI) 已经发生并正受到服务的时候才是可访问的。SMM、SMRAM 和 SMI 在 Intel 公司提供的微处理器手册中有描述。SMI 可以用与芯片集 123 外部 SMI 引脚连接的外部计时器 122 (图 2) 来生成。芯片集 123 用一个逻辑“或”函数 (未予示出) 将 SMI 与它的内部 SMI 源组合，然后将这个信号传出，输出到微处理器 SMI 引脚中。

或者，芯片集可含有用于电源管理的内部计时器或可编程成定期 SMI 的源的备用计时器。最好用这种内部计时器来实现，因为无需配备另外的硬件。计时器按编程的间隔上生成 SMI905。对于每个计时器事件，在 SMRAM 中存储的事件计数器增量 910。如果事件计数器达到预定值 915，说明操作系统已经崩溃，于是就启动数据收集。如果收集计数器还没有达到预定值，SMI 例程就被退出 935。一个在操作系统 901 控制任务下运行的程序定期地递减计数器或将计数器置零

902, 以防止计数器达到预定值。

监视计数器功能的实现细节必然依照其部署所在的是什么硬件平台而有所不同。用不同方式处理计数的其它等价实现也是可能的。

要是能够运行同一个程序来复位运行着相同操作系统的任何机器上的监视计时器程序—无论 SMRAM 中监视机制的实现细节如何, 那是有益的。因此, 常规 BIOS 500 最好配备一个 API 来启动监视器、关闭监视器、设置超时期限、复位 SMI 事件的当前计数。这个 API 最好以一个通过 BIOS 32 位服务目录访问的服务来实现。描述 BIOS 32 位服务的有《Standard BIOS 32-Bit Service Proposal》(0.4 版, Phoenix Technologies 出版, 1993 年 5 月 24 日) 一本文引以参考。在操作系统控制下运行的程序用具有内核访问权的设备驱动程序来访问 BIOS 32 位服务。

当这个推荐的监视计时器方法检测到操作系统崩溃时, 可以采取各种步骤 920 来为以后的诊断保留关于崩溃状态的信息。其中的例子是, 将计算机内存的一部分或全部, 复制到磁盘存储介质的预留区中—这通常称为核心转储; 保留屏幕存储器和视频控制器寄存器的状态—由此保留原本会由操作系统在其错误例程中显示的诊断信息。

在采取了所有这些保留证据的步骤之后, 系统可以被复位 930。这个复位在其 POST 的早期用一种在 CMOS 存储器中存储的特殊关机代码或其它等价装置向常规 BIOS 通知: 复位是崩溃所致。这个事实可以在控制台上显示, 并且如果网络增强型 BIOS 是这样配置的, 它就能因为这种崩溃而自动进入特殊诊断模式。可以将携带这个事实和关于崩溃的其它任选信息的 SNMP 报警通过网络 300 发送到另一个计算机。远离计算机的系统管理员于是就能选择在试图重新加载计算机操作系统之前用采用网络增强型 BIOS 的传输机制的应用将诊断信息传输到他的计算机。

特殊诊断状态

如上所述, BIOS 最好包括一个特殊诊断状态, 在这个状态下, 操作系统的加载被禁止, 计算机在图 4 的操作系统加载 414 之前, 等待有进一步的干预被启动。这就便于诊断导致要再引导系统的错误。可以用远程控制台装置来迫使计算机进入这个特殊诊断状态。这可以通过使用上述的远程过程调用来实现。网络增强型 BIOS 也可以这样配

置，使得如果计算机中发生某种故障条件，就可自动进入特殊诊断状态。

5 远程工作站 200 上的远程控制台可以结合网络增强型计算机 400 编程，在网络增强型计算机 400 处于特殊诊断状态期间执行许多有用的功能，诸如：使网络增强型计算机 400 复位并重新执行 POST 例程；致使数据从网络增强型计算机 400 上的硬盘被读出并通过网络 300 传输到远程控制台；使数据被从远程工作站 200 通过网络 300 传输到网络增强型计算机 400 并写入计算机 400 上的硬盘存储器 160。远程控制台可以进一步使 BIOS 中保留的诸如 CMOS 设置、DMI、ESCD 和
10 BIOS 错误日志等数据被通过网络 300 传输到远程工作站 200，或从远程工作站传输并在网络增强型计算机 400 的适当存储器中修改。

BIOS 也可以含有一个过程，用于远程加载 BIOS 本身一部分，以便能从网络 300 上的另一个计算机—诸如服务器 100—下载 BIOS 的一部分标准功能，由此降低在计算机上存储 BIOS 代码所需非易失性
15 存储器的存储量。如果必须从第二台计算机 100 下载 BIOS 的标准功能的基本部分，诸如 BIOS 代码，则计算机 400 如果不适当连接到网络 300 就不起作用。因为网络连接最好是编码过的，这就确保，如果计算机被偷或移开其网络连接，它将不再完成 POST 的执行并由此将使其是不可服务的。这将对偷窃的重要防止手段。

20 此外，远程控制台还可以使数据被通过网络 300 传输并写入用于存放常规 BIOS 和网络增强型 BIOS 的非易失性存储器 125，由此更新某些或部分 BIOS，使得第一台计算机下一次复位时将使用 BIOS 的新版本或配置。

应当明白，在不偏离本发明的范围和精神的情况下，对于本领域的
25 熟练人员来说，显然可以实现各种其它改进。例如，尽管图 1 的标注 400 中所示的是指台式计算机的网络增强型计算机，它也可以是一个服务器，如图 1 的标注 100 所示。因此，权利要求的范围并不限于本文所作的说明或解释，相反，权利要求被视为包含本发明中所有具有专利新颖性的特征，包括本领域熟练人员认为等价的那些特征。

说明书附图

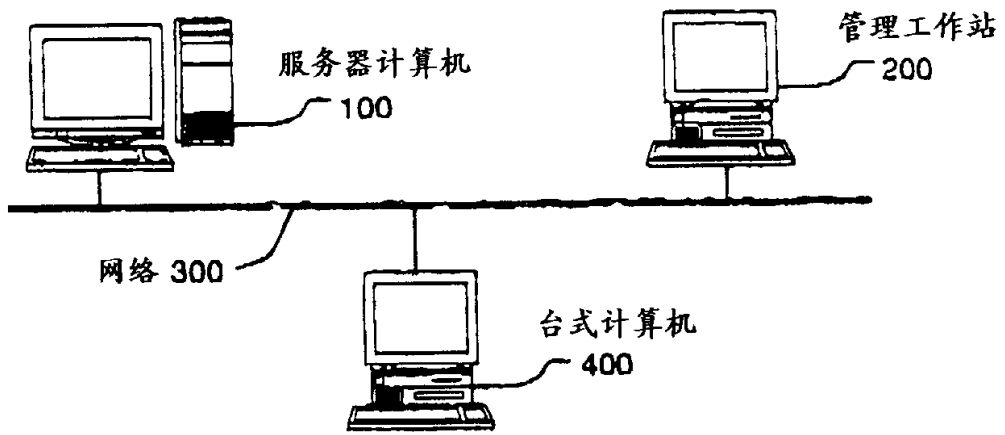


图 1

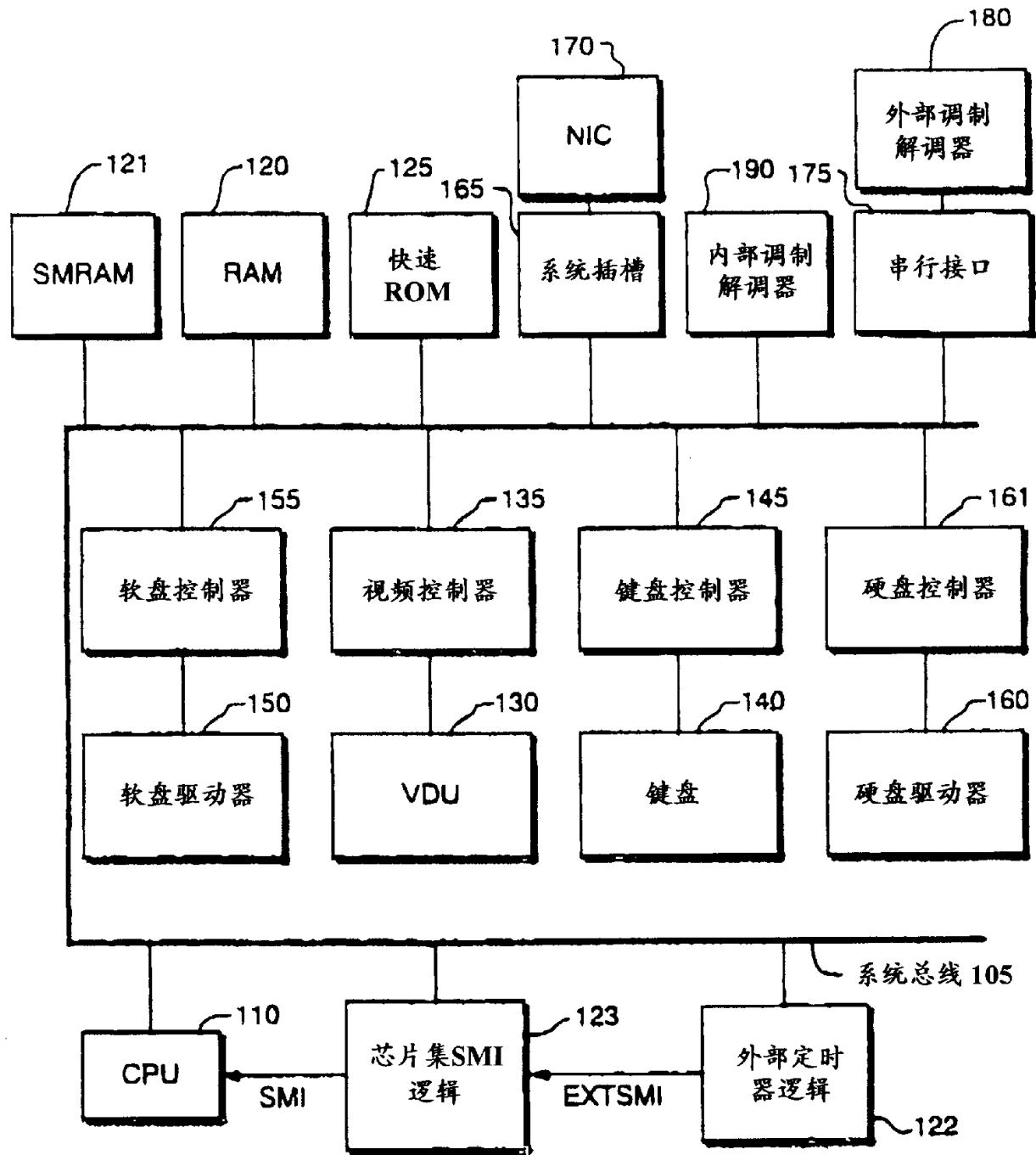


图 2

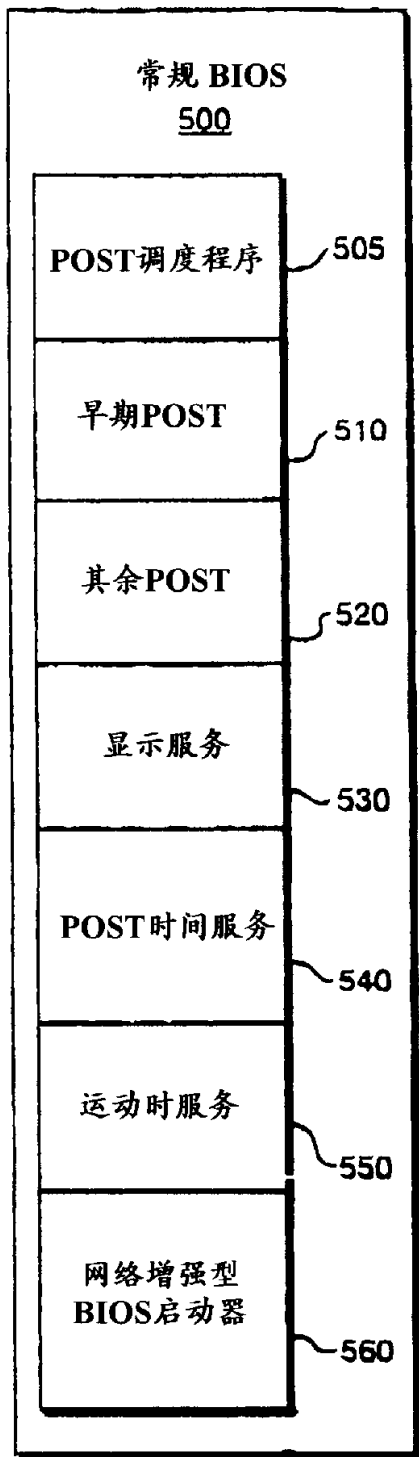


图 3A

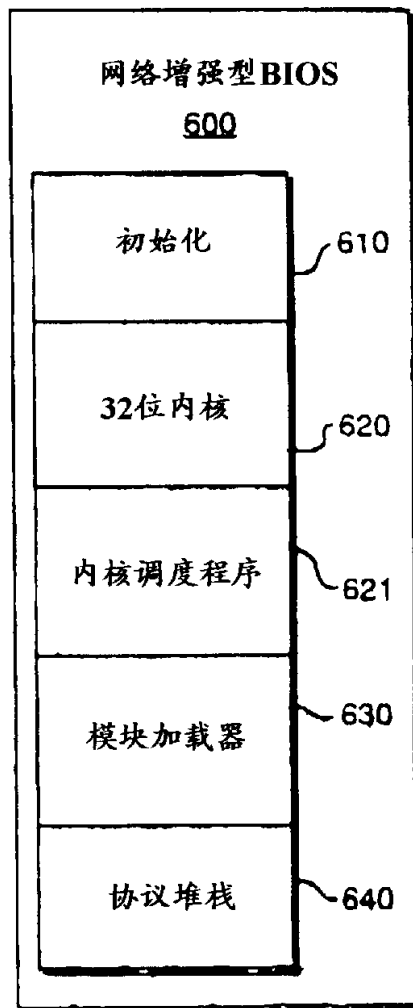


图 3B

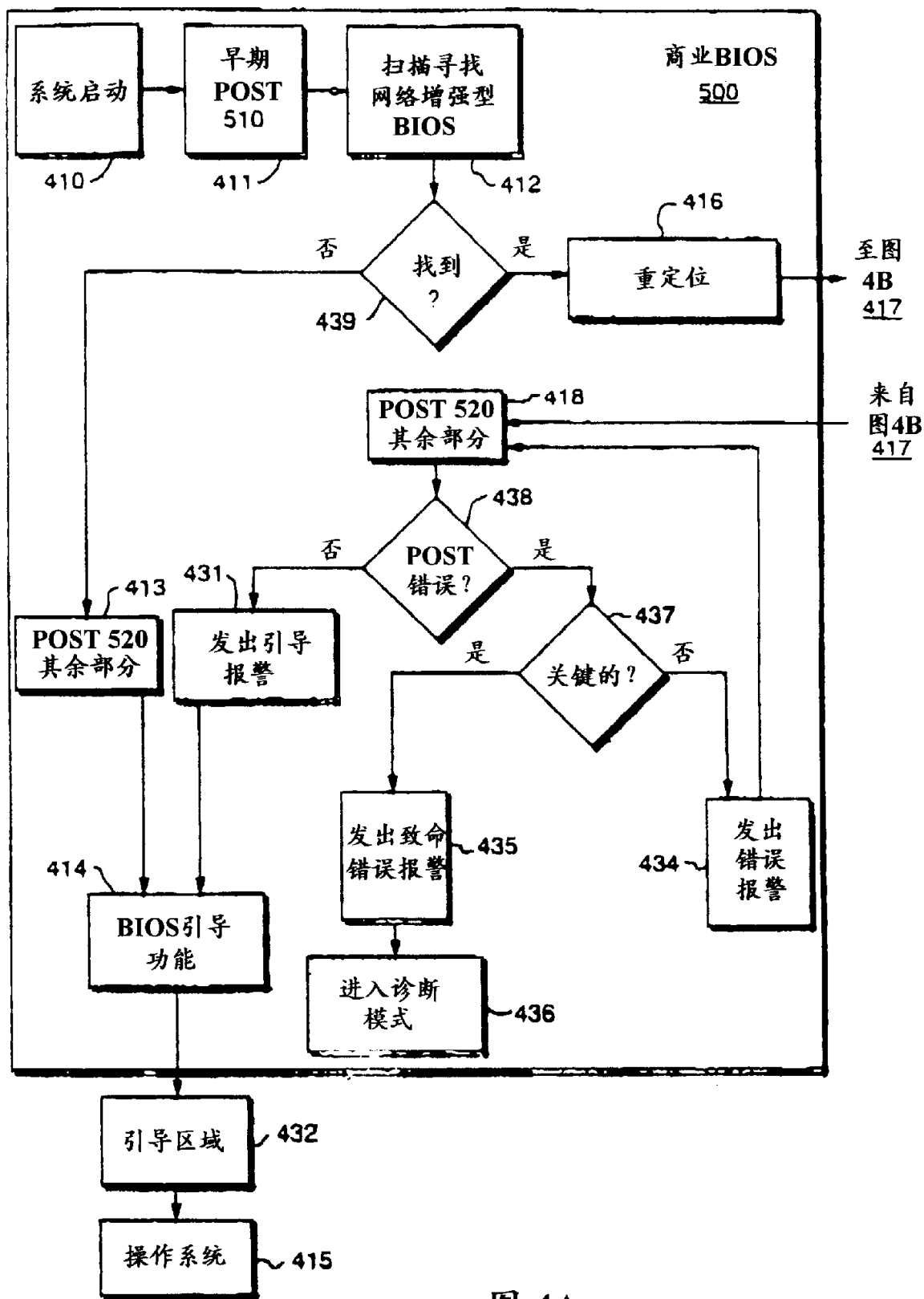


图 4A

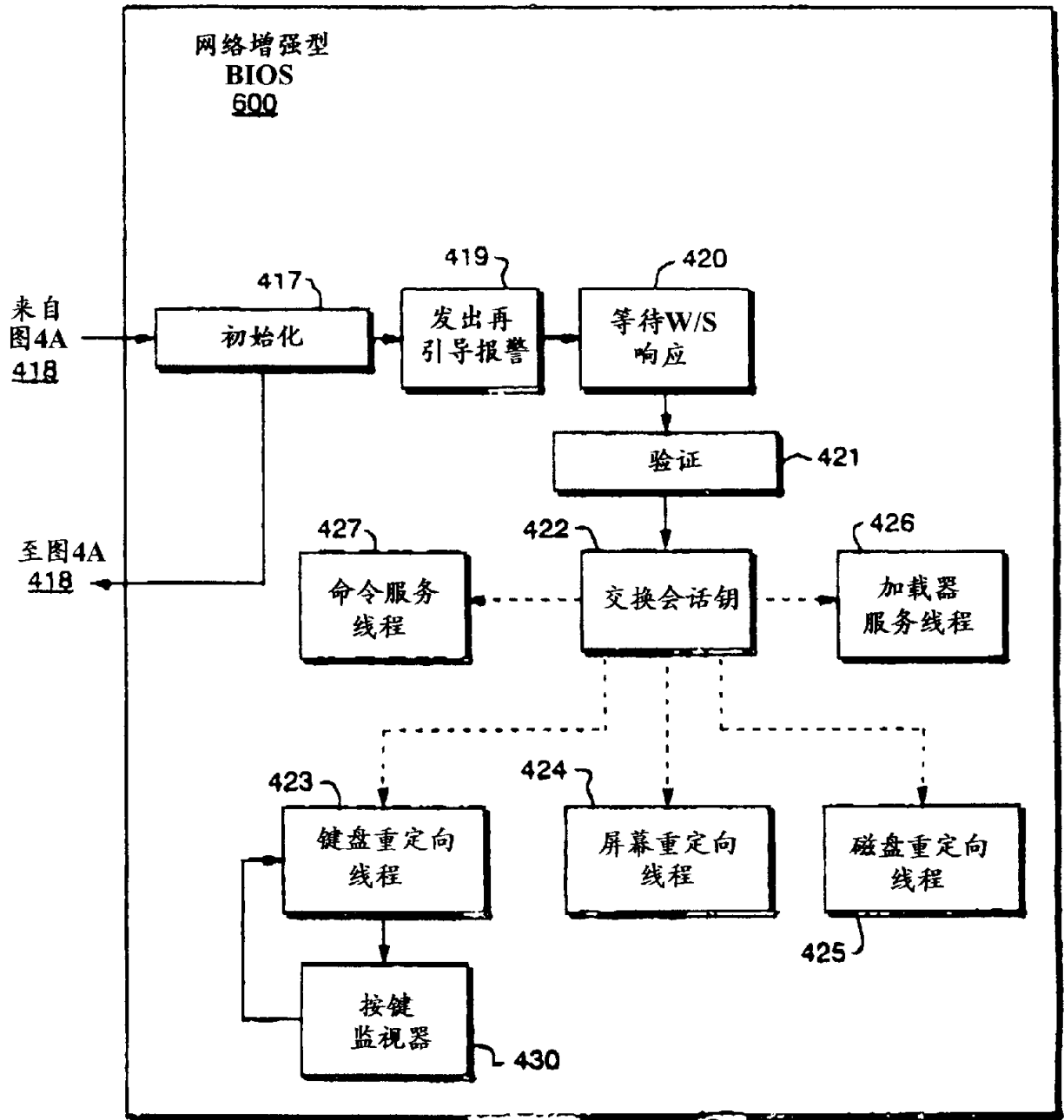


图 4B

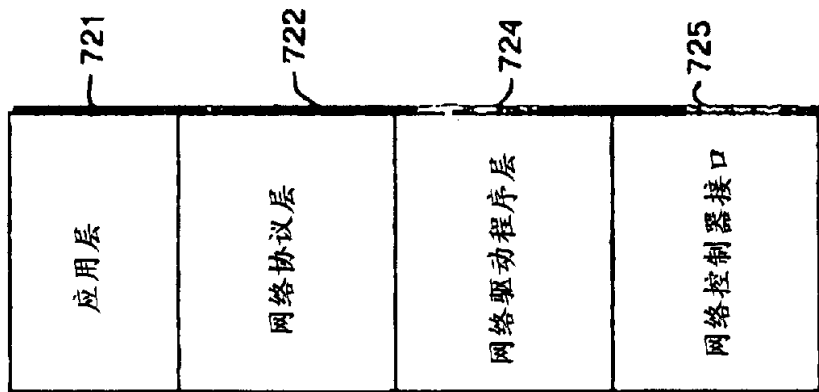


图 5

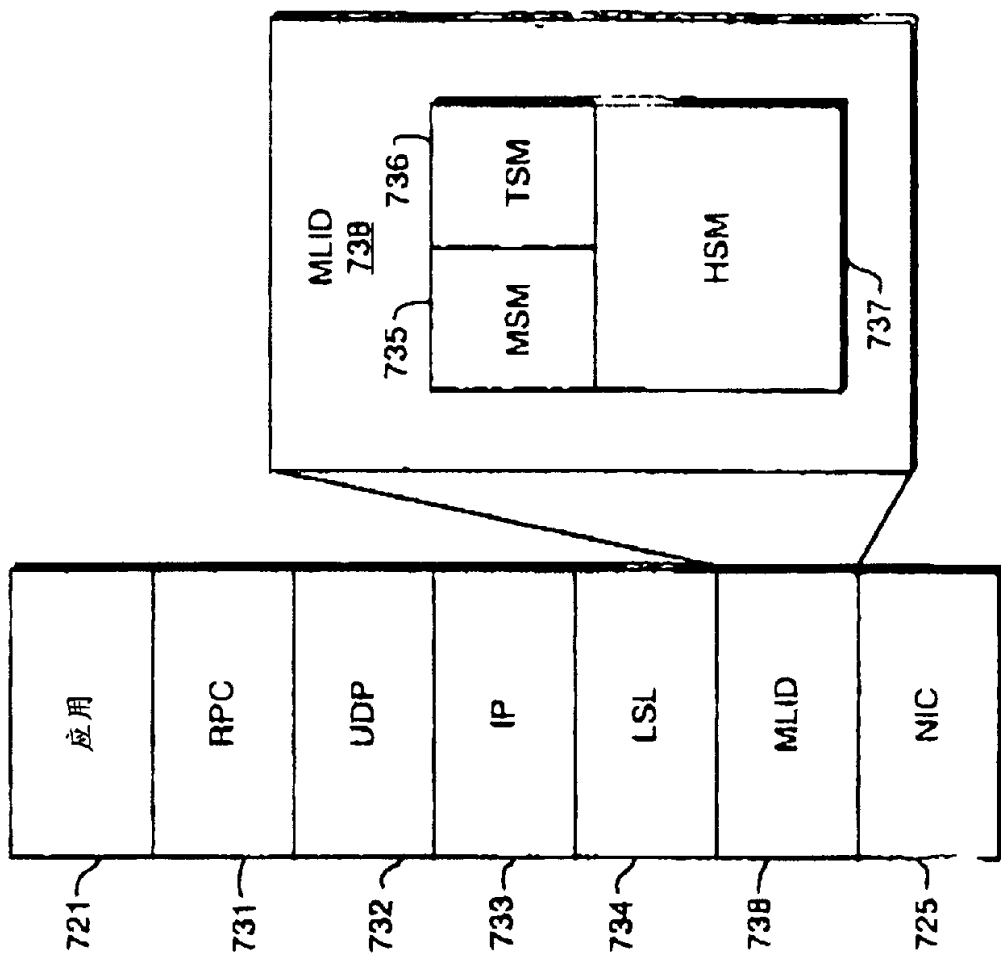


图 6

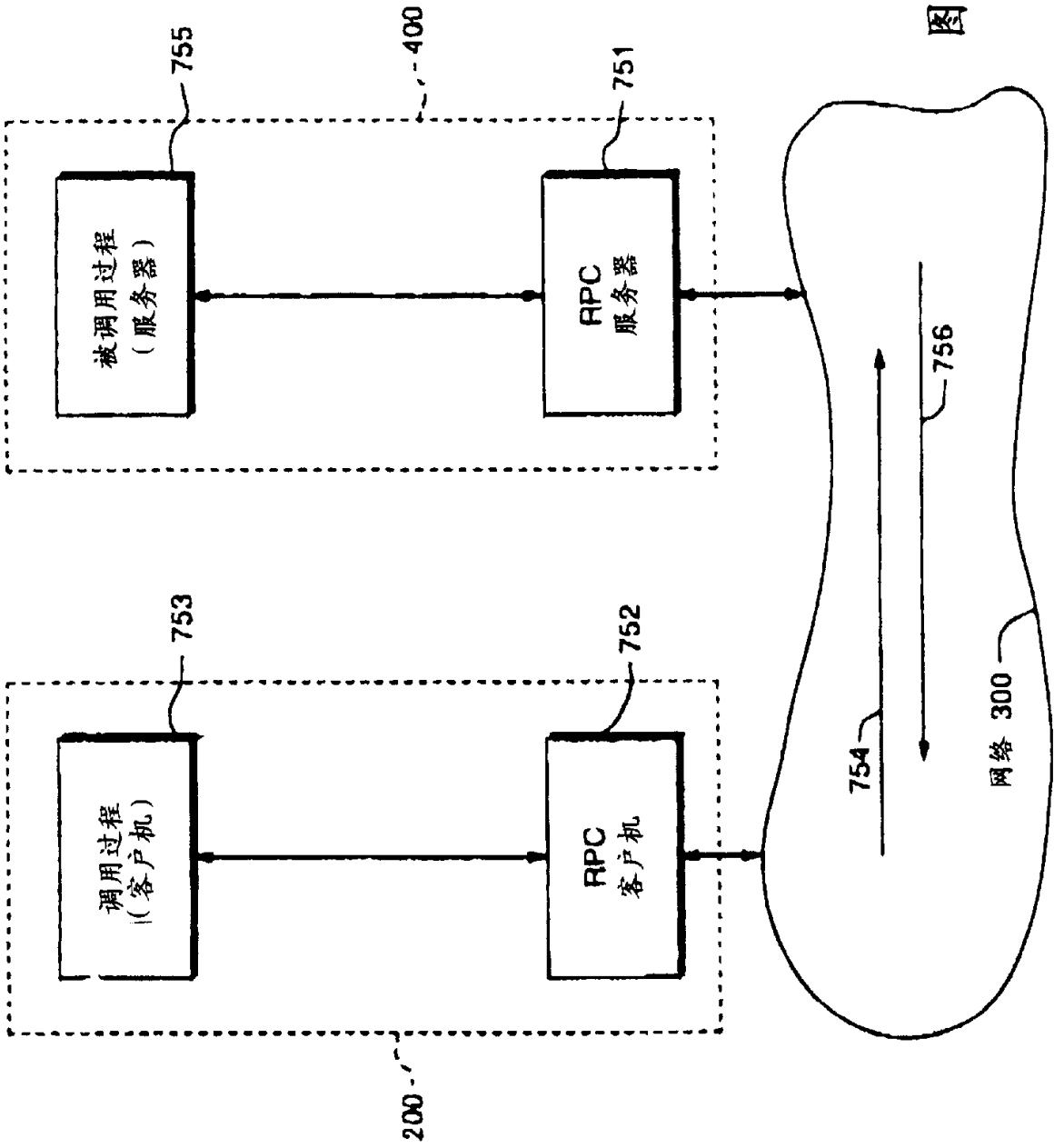


图 7

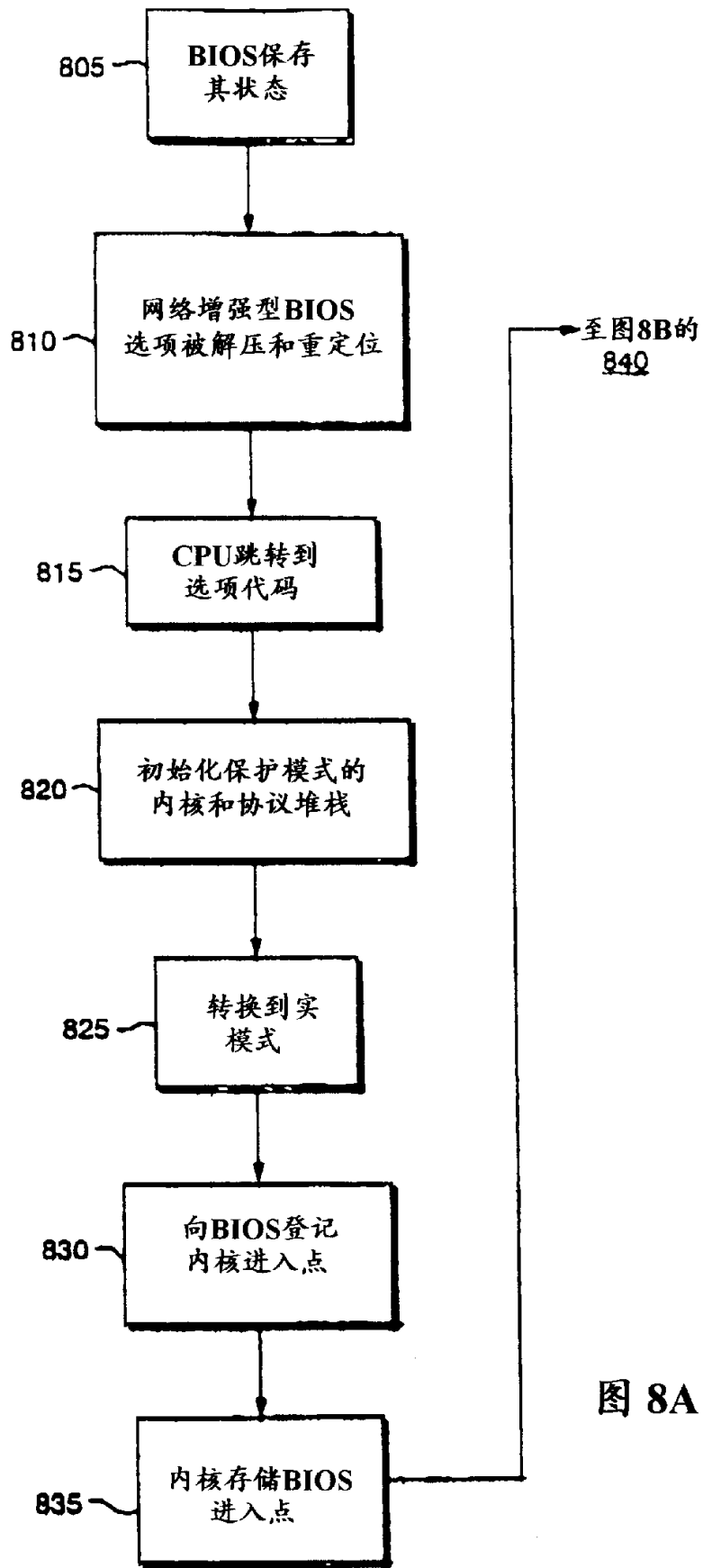


图 8A

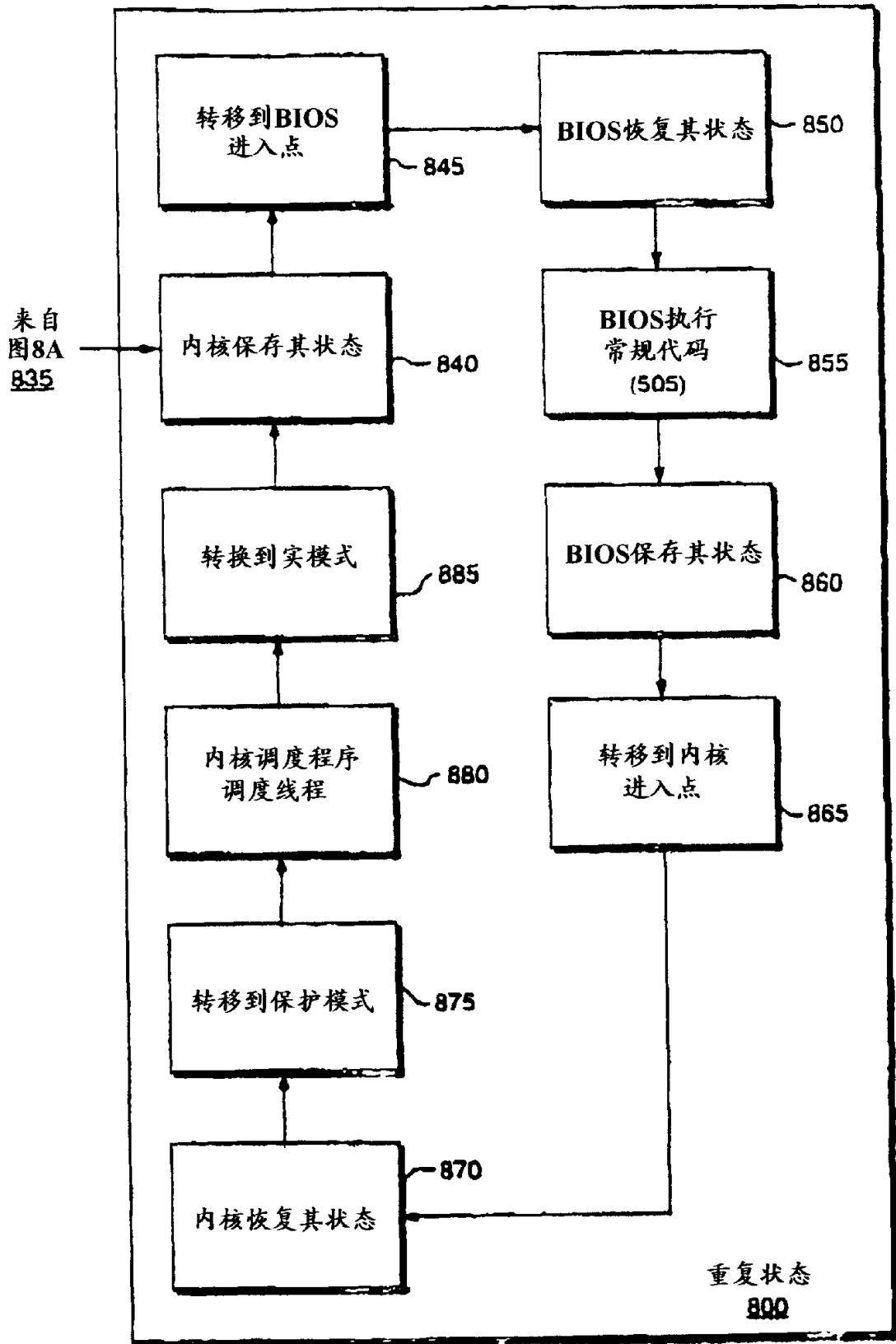


图 8B

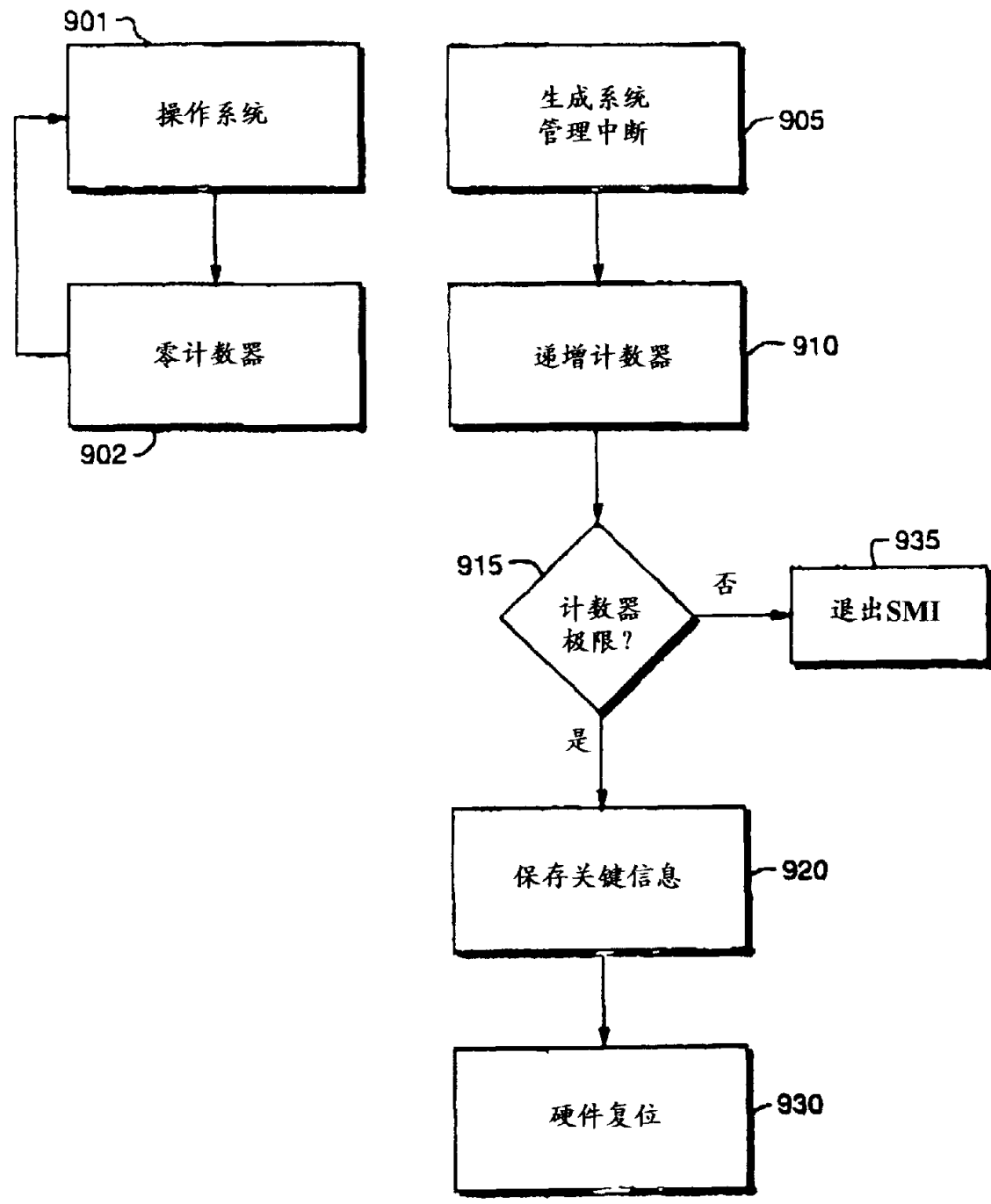


图 9