



[12] 发明专利申请公开说明书

[21] 申请号 200480003945.3

[43] 公开日 2006年8月9日

[11] 公开号 CN 1816790A

[22] 申请日 2004.1.12

[21] 申请号 200480003945.3

[30] 优先权

[32] 2003. 2. 14 [33] US [31] 10/367,566

[86] 国际申请 PCT/US2004/000726 2004. 1. 12

[87] 国际公布 WO2004/075034 英 2004. 9. 2

[85] 进入国家阶段日期 2005. 8. 11

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 J·卡尔达奇 B·贝尔蒙特

M·库马 R·杰克逊

G·丹尼尔斯 R·福兰德

V·古普塔 J·胡金斯

K·福莱明 U·加达姆塞蒂

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 钱慰民

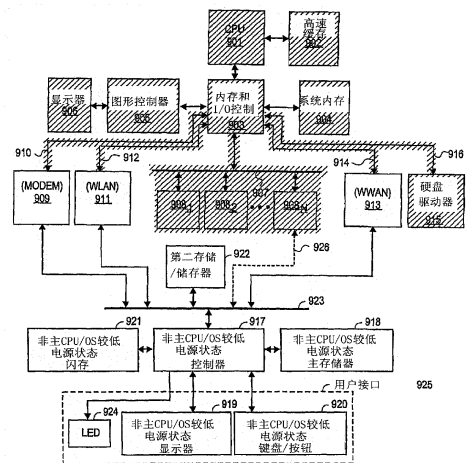
权利要求书9页 说明书22页 附图13页

[54] 发明名称

基于非主 CPU/OS 的操作环境

[57] 摘要

描述了一种计算机系统，其中包括一个 I/O 单元接口，当所述计算机系统在一个基于非主 CPU/OS 操作状态中操作时，它被取消活动。该计算机系统还包括一个控制器，当该计算机系统在一个基于非主 CPU/OS 操作状态中时，它执行功能性任务。该计算机系统还包括连接到该 I/O 单元接口和该控制器两者的一个 I/O 单元。



1. 一个计算系统，其特征在于，包含：
 - a) 一个 I/O 单元接口，当所述计算系统在一个基于非主 CPU/OS 操作状态中操作时，它被取消活动；
 - b) 一个控制器，当所述计算系统是在所述基于非主 CPU/OS 操作状态中时，它操作功能任务；和
 - c) 一个 I/O 单元，连接到所述 I/O 单元接口和所述控制器这两者。
2. 如权利要求 1 的计算系统，其中所述控制器进一步包含一个微处理器。
3. 如权利要求 1 的计算系统，其中所述控制器进一步包含一个微控制器。
4. 如权利要求 1 的计算系统，其中所述控制器作为一个逻辑状态机器被实现。
5. 如权利要求 1 的计算系统，其中所述控制器被连接到一个随机访问存储器。
6. 如权利要求 1 的计算系统，其中所述计算系统是一台笔记本或膝上型计算机。
7. 如权利要求 6 的计算系统，进一步包含一个 LED，它连接到所述控制器，当所述笔记本或膝上型计算机的盖子关闭时，所述 LED 对所述计算系统的用户是可见的。
8. 如权利要求 6 的计算系统，进一步包含一个显示器，它连接到所述控制器，当所述笔记本或膝上型计算机的盖子关闭时，所述显示器对所述计算系统的用户是可见的。
9. 如权利要求 6 的计算系统，进一步包含连接到所述控制器的一个键盘和/或一个或多个按钮，当所述笔记本或膝上型计算机的盖子关闭时，所述键盘和/或一个或多个按钮对所述计算系统的用户是可见的。
10. 如权利要求 1 的计算系统，进一步包含连接到所述控制器的一个 LED，当所述计算系统的主显示器对所述计算系统的用户是不可见时，所述 LED 对所述计算系统的用户是可见的。

11. 如权利要求 1 的计算系统,进一步包含连接到所述控制器的一个显示器,当所述计算系统的主显示器对所述计算系统的用户是不可见时,所述显示器对所述计算系统的用户是可见的。

12. 如权利要求 1 的计算系统,进一步包含连接到所述控制器的一个键盘和/或一个或多个按钮,当所述计算系统的主显示器对所述计算系统的用户是不可见时,所述键盘和/或一个或多个按钮对所述计算系统的用户是可见的。

13. 如权利要求 1 的计算系统,其中所述 I/O 单元通过一个总线连接到所述控制器。

14. 如权利要求 1 的计算系统,其中所述控制器进一步包含一个接口,在此数据到/从一个 FLASH 存储器能够被传播。

15. 如权利要求 1 的计算系统,其中所述控制器进一步包含一个接口,在此数据到/从一个聚合物存储器能够被传播。

16. 如权利要求 1 的计算系统,其中所述 I/O 单元接口是一个串行端口。

17. 如权利要求 16 的计算系统,其中所述 I/O 单元是一个 MODEM。

18. 如权利要求 17 的计算系统,其中所述 MODEM 是一个 V.90 MODEM。

19. 如权利要求 1 的计算系统,其中所述 I/O 单元接口是一个 USB 接口。

20. 如权利要求 19 的计算系统,其中所述 I/O 单元是一个基于 IEEE 802.11 的网络接口或者一个基于 BLUETOOTH 的网络接口。

21. 如权利要求 1 的计算系统,其中所述 I/O 单元接口是一个用于无线 MODEM 的串行接口。

22. 如权利要求 21 的计算系统,其中所述无线 MODEM 是一个 GPRS MODEM。

23. 如权利要求 1 的计算系统,其中所述 I/O 单元是一个无线 LAN I/O 单元。

24. 如权利要求 1 的计算系统,其中所述 I/O 单元是一个无线 WAN I/O 单元。

25. 如权利要求 1 的计算系统,其中所述 I/O 单元接口是一个对所述计算系统的主系统总线的接口。

26. 如权利要求 1 的计算系统，其中所述主系统总线是一个 PCI 总线。

27. 如权利要求 1 的计算系统进一步包含一个操作系统（OS），它是在所述控制器上执行，所述 OS 包含下述部件：

a) 一个应用程序程序员接口（API）；

b) 一个用户接口管理功能，当所述计算系统处于所述基于非主 CPU/OS 操作状态时，它控制运作的用户接口；和

c) 一个储存管理功能，当所述计算系统处于所述基于非主 CPU/OS 操作状态时，它管理运作的的数据储存资源。

28. 如权利要求 27 的计算系统进一步包含应用软件，当所述计算系统处于所述基于非主 CPU/OS 操作状态时，它通过所述 API 运行在所述 OS 上。

29. 如权利要求 28 的计算系统，其中所述应用软件进一步包含数据存储应用软件程序。

30. 如权利要求 29 的计算系统，其中所述数据存储应用软件程序被配置为：

a) 从所述 I/O 单元向存储数据对象的服务器启动一个无线通信；和

b) 在所述数据对象由所述服务器发送到所述计算系统并且由所述计算系统接收之后，存储所述数据对象到所述数据存储资源中。

31. 如权利要求 30 的计算系统，其中所述无线通信是一个 WWAN 通信。

32. 如权利要求 30 的计算系统，其中所述无线通信是一个 WLAN 通信。

33. 如权利要求 30 的计算系统，其中所述数据对象进一步包含日历信息，供所述用户接口使用。

34. 如权利要求 30 的计算系统，其中所述数据对象进一步包含联系信息，供所述用户接口使用。

35. 如权利要求 30 的计算系统，其中所述数据对象进一步包含“到-做”信息，供所述用户接口使用。

36. 如权利要求 28 的计算系统，其中所述应用软件进一步包含一个功能应用软件程序。

37. 如权利要求 36 的计算系统，其中所述功能是一个音乐回放功能，所述功能应用软件程序配置为：

a) 从所述数据存储资源恢复编码的音乐；和，

b) 解码所述编码的音乐。

38. 如权利要求 37 的计算系统, 其中所述编码的音乐是在一种 MP3 格式中的编码。

39. 如权利要求 37 的计算系统, 其中所述功能应用程序进一步被配置, 以启动所述音乐在它的解码之后传输到无线头戴耳机。

40. 如权利要求 37 的计算系统, 其中所述用户接口, 当所述系统处于所述基于非主 CPU/OS 操作状态时, 提供用户选择歌曲的能力, 对此所述音乐至少是一个部分。

41. 如权利要求 37 的计算系统, 其中所述用户接口, 当所述系统处于所述基于非主 CPU/OS 操作状态时, 提供用户控制所述音乐的音量的能力。

42. 如权利要求 27 的计算系统, 其中所述用户接口管理功能能够响应于由用户按动的按钮, 把数据对象呈现在一个显示器上。

43. 如权利要求 42 的计算系统, 其中所述数据对象包括一个嵌入的导航命令, 它描述将被显示在所述显示器上的所述数据对象在哪里。

44. 如权利要求 43 的计算系统, 其中所述计算系统是一个膝上型或笔记本计算机, 并且当所述膝上型或笔记本计算机的盖子是关闭时, 其中所述显示器也能够被用户观察。

45. 如权利要求 27 的计算系统, 其中所述 API 能够接收从一个应用程序发送来的数据对象, 当所述计算系统不处于所述基于非主 CPU/OS 操作状态中并且当计算系统是处于或正常开机或低电源基于主 CPU/OS 操作状态中时, 该程序运行在所述计算系统的主 OS 上。

46. 如权利要求 45 的计算系统, 其中所述数据对象管理功能能够在所述数据对象由所述 API 接收之后, 存储所述数据对象到所述存储资源之中。

47. 如权利要求 45 的计算系统, 其中所述用户接口管理功能能够在所述数据对象由所述 API 接收之后, 把所述数据对象呈现到所述用户接口上。

48. 一个计算系统, 其特征在于, 包含:

a) 一个主系统总线, 当所述计算系统操作处于基于非主 CPU/OS 操作状态时, 它保持活动;

b) 一个控制器, 当所述计算系统处于所述基于非主 CPU/OS 操作状态中时,

它操作功能任务，所述控制器连接到所述主系统总线；和，

c) 一个 I/O 单元，当所述计算系统操作处于所述基于非主 CPU/OS 操作状态中时，它连接到保持活动的所述主系统总线。

49. 如权利要求 48 的计算系统，其中所述控制器进一步包含一个微处理器。

50. 如权利要求 48 的计算系统，其中所述控制器进一步包含一个微控制器。

51. 如权利要求 48 的计算系统，其中所述控制器作为一个逻辑状态机器被实现。

52. 如权利要求 48 的计算系统，其中所述控制器被连接到一个随机访问存储器。

53. 如权利要求 48 的计算系统，其中所述计算系统是一台笔记本或膝上型计算机。

54. 如权利要求 53 的计算系统，进一步包含一个 LED，它连接到所述控制器，当所述笔记本或膝上型计算机的盖子关闭时，所述 LED 对所述计算系统的用户是可见的。

55. 如权利要求 53 的计算系统，进一步包含一个显示器，它连接到所述控制器，当所述笔记本或膝上型计算机的盖子关闭时，所述显示器对所述计算系统的用户是可见的。

56. 如权利要求 55 的计算系统，其中所述显示器是装配在所述盖子的背后。

57. 如权利要求 55 的计算系统，其中所述显示器是装配在所述计算系统的边上，而不是所述盖子的背后。

58. 如权利要求 53 的计算系统，进一步包含连接到所述控制器的一个键盘和/或一个或多个按钮，当所述笔记本或膝上型计算机的盖子关闭时，所述一个键盘和/或一个或多个按钮对所述计算系统的用户是可见的。

59. 如权利要求 48 的计算系统，进一步包含连接到所述控制器的一个 LED，当所述计算系统的主显示器对所述计算系统的用户是不可见时，所述 LED 对所述计算系统的用户是可见的。

60. 如权利要求 48 的计算系统，进一步包含连接到所述控制器的一个显示器，当所述计算系统的主显示器对所述计算系统的用户是不可见时，所述显示

器对所述计算系统的用户是可见的。

61. 如权利要求 48 的计算系统, 进一步包含连接到所述控制器的一个键盘和/或一个或多个按钮, 当所述计算系统的主显示器对所述计算系统的用户是不可见时, 所述键盘和/或一个或多个按钮对所述计算系统的用户是可见的。

62. 如权利要求 48 的计算系统, 其中所述主系统总线是一个 PCI 总线。

63. 一种机器可读介质, 其特征在于, 包含:

使得控制器实现操作系统的指令, 所述操作系统和所述控制器正在运作, 而一个计算系统的主 CPU 正在睡眠, 所述操作系统包含;

a) 一个应用程序程序员接口 (API);

b) 一个用户接口管理功能, 当所述计算系统的主显示器是非活动时, 它控制运作的用户接口; 和

c) 一个储存管理功能, 当所述计算系统的主 CPU 是睡眠时, 它管理运作的的数据储存资源。

64. 如权利要求 63 的机器可读介质, 其中所述控制器是一个微处理器。

65. 如权利要求 63 的机器可读介质, 其中所述控制器是一个微控制器。

66. 如权利要求 63 的机器可读介质, 其中所述用户接口包含一个显示器。

67. 如权利要求 63 的机器可读介质, 其中所述用户接口进一步包含一个或多个按钮。

68. 如权利要求 67 的机器可读介质, 其中所述用户接口进一步包含一个键盘。

69. 如权利要求 63 的机器可读介质, 其中所述用户接口进一步包含一个 LED。

70. 一种机器可读介质, 其特征在于, 包含:

指令, 当由活动的控制器执行时, 而一个计算系统的主 CPU 正在睡眠, 使得所述控制器实行一种方法, 所述方法包含;

a) 从所述 I/O 单元向存储数据对象的服务器启动一个无线通信; 和

b) 在所述数据对象由所述服务器发送到所述计算系统并且由所述计算系统接收之后, 存储所述数据对象到数据存储资源中, 所述 I/O 单元和所述数据存储资源是活动的, 而所述计算系统的主 CPU 正在睡眠。

71. 如权利要求 70 的机器可读介质, 其中所述无线通信是一个 WWAN 通信。

72. 如权利要求 70 的机器可读介质, 其中所述无线通信是一个 WLAN 通信。

73. 如权利要求 70 的机器可读介质, 其中所述数据对象进一步包含日历信息供用户接口使用, 该接口是活动的, 而所述计算系统的主 CPU 正在睡眠。

74. 如权利要求 70 的机器可读介质, 其中所述数据对象进一步包含联系信息供用户接口使用, 该接口是活动的, 而所述计算系统的主 CPU 正在睡眠。

75. 如权利要求 70 的机器可读介质, 其中所述数据对象进一步包含“到-做”信息供用户接口使用, 该接口是活动的, 而所述计算系统的主 CPU 正在睡眠。

76. 如权利要求 70 的机器可读介质, 其中所述控制器是一个微处理器。

77. 如权利要求 70 的机器可读介质, 其中所述控制器是一个微控制器。

78. 一种机器可读介质, 其特征在于, 包含:

指令, 当由活动的控制器执行而一个计算系统的主 CPU 正在睡眠时, 使得所述控制器实行一种方法, 所述方法包含;

a) 从活动的数据存储资源恢复编码的音乐, 而所述计算系统的主 CPU 正在睡眠; 和,

b) 解码所述编码的音乐。

79. 如权利要求 78 的机器可读介质, 其中所述编码的音乐是在一种 MP3 格式中的编码。

80. 如权利要求 78 的机器可读介质, 其中所述方法进一步包含启动所述音乐在它的解码之后的传输到一个头戴耳机。

81. 如权利要求 78 的机器可读介质, 其中所述方法进一步包含允许用户选择歌曲, 所述音乐至少是所述歌曲的一部分。

82. 如权利要求 78 的机器可读介质, 其中所述方法进一步包含允许用户控制所述音乐的音量。

83. 如权利要求 82 的机器可读介质, 其中所述控制器是一个微处理器。

84. 如权利要求 82 的机器可读介质, 其中所述控制器是一个微控制器。

85. 一种方法，其特征在于，包含：

闪烁安装在计算系统上的 LED，当所述计算系统的主显示器对所述计算系统的用户不可见时，所述 LED 对用户是可见的，当所述计算系统被操作处于一个基于主 CPU/OS 低电源状态中时，所述 LED 由一个事件的到来而促使成为闪烁，所述闪烁由一个控制器控制，当所述计算系统操作处于一个基于非主 CPU/OS 较低电源状态中时，该控制器保持活动。

86. 如权利要求 85 的方法，其中所述 LED 的所述闪烁是通知所述用户从一个未知呼叫来的记录信息正等待被听取，所述事件是所述消息的记录，其中所述呼叫没有被应答的判断是从所述基于非主 CPU/OS 较低电源状态中作出。

87. 由一个计算系统实行的一种方法，其特征在于，包含：

a) 虽然所述计算系统的主 CPU 正在睡眠，从所述计算系统启动从所述计算系统的 I/O 单元向存储有数据对象的一个服务器的一个无线通信；和，

b) 虽然所述计算系统的主 CPU 正在睡眠，在所述数据对象已经由所述服务器发送到所述计算系统并且由所述计算系统接收之后，存储所述数据对象到数据存储资源中。

88. 如权利要求 87 的方法，其中所述无线通信是一个 WWAN 通信。

89. 如权利要求 87 的方法，其中所述无线通信是一个 WLAN 通信。

90. 如权利要求 87 的方法，其中所述数据对象进一步包含日历信息供活动的用户接口使用，而所述计算系统的主 CPU 正在睡眠。

91. 如权利要求 87 的方法，其中所述数据对象进一步包含联系信息供活动的用户接口使用，而所述计算系统的主 CPU 正在睡眠。

92. 如权利要求 87 的方法，其中所述数据对象进一步包含“到-做”信息供活动的用户接口使用，而所述计算系统的主 CPU 正在睡眠。

93. 由一个计算系统实行的一种方法，其特征在于，包含：

a) 从活动的数据存储资源恢复编码的音乐，而所述计算系统的主 CPU 正在睡眠；和，

b) 解码所述编码的音乐，而所述计算系统的主 CPU 正在睡眠。

94. 如权利要求 93 的方法，其中所述编码的音乐用 MP3 格式编辑。

95. 如权利要求 93 的方法，其中所述方法进一步包含启动所述音乐在它的

解码之后的传输到一个头戴耳机。

96. 如权利要求 93 的方法，其中所述方法进一步包含允许用户选择歌曲，所述音乐至少是所述歌曲的一部分。

97. 如权利要求 93 的方法，其中所述方法进一步包含允许用户控制所述音乐的音量。

基于非主 CPU/OS 的操作环境

发明领域

本发明领域通常涉及计算，更具体地涉及基于非主 CPU/OS 的操作环境。

背景

A. 计算系统

图 1 表示一个计算系统 100 实施例。该计算系统包括一个中央处理单元 (CPU) 101，一个高速缓存 102，内存和 I/O 控制 103，和系统内存 104。由该计算系统执行的软件指令（和它相应的数据）被存储在系统内存 104 和高速缓存 102 中（在此经常使用的指令和数据被存储在高速缓存 102 中）。该软件指令（与相应的数据一起）由该 CPU 101 执行。主存的存储控制器部分和 I/O 控制 103 负责管理对系统内存 104 的访问（它可以被功能元件例如图形控制器 105 和各种 I/O 单元使用，而不只是 CPU101 使用）。

图形控制器 105 和显示器 106 提供由计算机系统 100 的用户观察到的计算机生成的图像。该 I/O 内存控制器和 I/O 控制功能 103 负责管理对系统内存 104 对各种 I/O 单元 108_1 到 108_N 和 109、111、113 和 115 的访问。I/O 单元典型地被看作为功能单元，它发送（接收）信息到（从）该计算系统（例如网络适配器、MODEM、无线接口、键盘、鼠标等）和/或用于存储在计算系统 100 中的该计算系统的信息的功能单元（例如，硬盘驱动器单元）。

各种 I/O 单元频繁地在计算系统中发现；并且进一步，各种类型用于在 I/O 单元和 I/O 控制功能之间通信的接口被频繁地在计算系统中发现。通常，这些接口由工业标准定义。图 1 的示例性计算系统体系结构表示一个系统总线接口 107，不同的 I/O 单元 108_1 到 108_N 可被插入在其中；以及，不同的接口 110、112、114 和 116。不同接口 110、112、114 和 116 的每一个在图 1 中被表示，作为具有它自己的相应的 I/O 单元 109、111、113 和 115。

注意，不同的计算系统可接受不同数量的接口；并且，不同的计算机系统

可接受不同的接口类型（例如，每个接口的 I/O 单元的最大数量、接口技术等方面）。仅仅作为一个可能的实现，使用图 1 的计算系统作为模板：1) 系统总线 107 是 PCI 总线；2) 接口 110 是串行端口；3) 接口 112 是 USB 接口；4) 接口 114 是串行接口；和 5) 接口 116 是 IDE 接口（或者其他存储设备接口）。

B. 计算系统状态图

图 2 表示一个现有技术中的计算系统技术状态图。在图 2 中看到的该操作状态实施例可在 Advanced Configuration and Power Interface (ACPI) 技术规范（修订版 2.0A，日期 2002 年 3 月 31 日）中找到（由康帕克计算机公司、英特尔公司、微软公司、凤凰技术公司和东芝公司发表）。虽然该 ACPI 技术规范被公认为描述大量已存在的计算系统，应该认识到大量不遵从 ACPI 技术规范的计算系统也仍然能够遵从于在图 2 中看到的操作状态的配置。因而，图 1 的描述相应于 ACPI 技术规范被遵守的更加普遍的描述。

根据图 2 的描述在第一状态 201，被称为“正常开”状态 201，是该计算机的正常操作状态（即，它被激活电源并正（或已准备好）被用户使用的状态）。在 ACPI 技术规范中，该“正常开”状态 201 被称作为“G0”状态。第二状态 202 表示任何一个或多个状态，其中该计算系统被识别为是正“关机（off）”。该 ACPI 技术规范识别两种这样的状态：基于硬件的关机状态（例如，其中电源已经从整个系统被移除）和基于软件的关机状态（其中电源被提供给系统但是 BIOS 和操作系统（OS）必须从残痕中被重载，而没有参照存储的先前操作周围的环境）。该 ACPI 技术规范称基于硬件的关机状态为“G3”状态和基于软件的关机状态为“G2”状态。

第三状态 203 是指任何一个或多个状态，其中该计算系统被识别为“睡眠（sleep）”。对于睡眠状态，在“正常开机”状态 201 中系统的操作环境（例如，各种软件例程的状态和数据）被预先保存到正在进入低功耗状态的计算机的 CPU 中。该睡眠状态 203 目标是在暂停期间节约 CPU 的功耗而继续使用该计算系统。例如，这就是如果用户正在正常开机状态 201 中使用一个计算系统（例如，打字输入文档）并且然后被打扰注意力而必须临时制止该使用（例如，接听一个电话）-该计算系统能够自动地从正常开机状态 201 转换到睡眠状态 202 以减少系统的能耗。

在此，该计算系统的软件操作环境（例如，包括被书写的文档），它也被

称作为“环境”或“该环境”，预先被保存。结果，在打扰结束之后当用户返回来使用该计算系统时，该计算系统能够自动地呈现给用户打扰发生时存在的环境（通过重调用被保存的环境）作为从睡眠状态 203 转换回到正常状态 201 的一部分。该 ACPI 技术规范识别不同睡眠状态的集合（可表示为“S1”、“S2”、“S3”和“S4”状态），每一个都各自具有它自己的当返回到“正常开机”状态 201 时电源节省和延迟之间的平衡（这里，S1、S2、和 S3 状态被识别为“备用（standby）”的各个口味，而 S4 状态是“冬眠（hibernate）”状态。

然而，先前技术睡眠状态的问题是 CPU 不能执行任何有用的工作。这样，尽管电源节省被识别，任何在计算系统睡眠期间可以被有用地执行的任务都不可能实现。

附图简述

参照被用于表示本发明实施例的以下说明和相关的附图，可以更好地理解本发明。在附图中：

图 1 展示一个计算系统的实施例；

图 2 展示对于计算系统的现有技术状态图；

图 3 展示具有有用的低电源状态的计算系统的改进状态图；

图 4a 到图 4c 分别演示对于“正常开机”状态（图 4a）、“基于主 CPU/OS 低电源”状态（图 4b）、和“基于非主 CPU/OS 较低电源”状态（图 4c）的活动的和非活动的计算系统硬件组件之间关系的实施例；

图 5 展示为各自地跨越“正常开机”状态、“基于主 CPU/OS 低电源”状态、和“基于非主 CPU/OS 较低电源”状态而完整的电话系统的各种功能角色分配的实施例；

图 6a 和 6b 演示示例性方法流程，用于从“正常开机”状态到“基于主 CPU/OS 低电源”状态的转换（图 6a）；和用于从“基于主 CPU/OS 低电源”状态到“正常开机”状态的转换（图 6b）；

图 7 展示状态转换逻辑的实施例，该逻辑可被用来帮助/控制计算系统状态之间的转换：1) 正常开机状态和一个或多个睡眠状态之间；和 2) “基于主 CPU/OS 低电源”状态和“基于非主 CPU/OS 较低电源”状态之间；

图 8a 和 8b 演示示例性方法流程，用于从“基于主 CPU/OS 低电源”状态到“基于非主 CPU/OS 较低电源”状态的转换（图 8a）；和用于从“基于非主

CPU/OS 较低电源”状态到“基于主 CPU/OS 低电源”状态的转换（图 8b）；

图 9a 展示一个更详细的计算系统的实施例，该系统具有低电源，但是可操作基于非主 CPU/OS 子系统；

图 9b 展示一对移动计算系统的实施例，其每一个提供一个“关闭盖”用户接口；

图 10 展示用于“基于主 CPU/OS 低电源”状态的软件体系结构的实施例，用于“基于非主 CPU/OS 较低电源”状态的软件体系结构的实施例；和可能的这一对状态之间的关系；

图 11 表示若干种状态转换的例子，它们随时间重新作用使用计算系统。

说明

具有操作性低电源状态的状态图和计算系统

为了使计算系统在低电源功耗状态下执行有用的任务，特定的状态必须被设计到系统中。尤其是，这些特定的状态应该被配置具有足够数量的功能本领，使得一个或多个有用的任务能够被执行；而且在同时，按关联于“正常开机”状态较为低的比例节约电源。这里，图 3 和图 4a 到 4c 表示具有两个这种特定状态的计算系统的实施例。

图 3 呈现一个状态图。图 4a 到 4c 展示示例性计算系统的各种部件的示例性描述，该计算系统并不被强制地进入低电源状态，或者被强制地进入低电源状态而其中从每个状态有用的任务可以被执行（这里阴影区域表示部件被强制保留在非活动低电源状态，而非阴影区域表示部件没有被强制保留在非活动低电源状态）。重要的是注意如在以下将详细讨论的那样，在图 4a 到图 4c 看到的非活动低电源状态被强制的部件和不是非活动低电源状态被强制的部件的计算机系统和特殊的组合两者都是示例性的，并且可以随着实施例的不同而不同。

根据图 3 所示的状态图方案，一个能够执行有用任务的计算系统具有 3 个主要的状态：1) 一个高电源，“正常开机”状态 301；2) 一个“基于主 CPU/OS 低电源”状态 304；和 3) 一个“非基于主 CPU/OS 较低电源”状态 305。图 4a 到图 4c 展示用于以上描述状态的每一个的单个计算机系统示例实施例。这些状态的每一个简要概述以下立即描述，随后是更全面的“主 CPU/OS 低电源”状态 304 和“非主 CPU/OS 较低电源”状态 305 的讨论。

图 4a 展示一个“正常开机”状态 301 计算系统的实施例。注意计算系统的各种部件都没有被强制地进入非活动低电源状态，因为部件都不是阴影的。在各种实施例中，至少某些计算系统的部件被给予权力根据检测到的用途调整它们自身的电源消耗（例如，从最低电源消耗状态到最高电源消耗状态）。这里，通过不强制部件进入非活动低电源状态，具有能力调整它们自身的电源消耗的部件就在该“正常开机”状态 301 中自由地实行调整。

对比之下，图 4b 展示同样的计算系统在它被放置到“基于主 CPU/OS 低电源”状态 304 之后的实施例。这里，特定部件（由阴影指出）已经被强制进入非活动低电源状态。与这种观点相一致，具有权力调整它们自身的电源消耗的那些部件并且将被强制地进入非活动低电源状态的部件被剥夺它们的电源调整权力并且强制地进入它们的最低电源消耗状态。这里，注意术语“非活动”指该部件停止了它的主功能从而电源可以被节省。注意，该主 CPU 没有被非活动化并且因此能够继续执行软件程序。

图 4c 展示同样的计算系统在它被放置到“基于非主 CPU/OS 较低电源”状态 305 之后的实施例。这里，注意附加部件（特别是，CPU）已经如由与图 4b 相比较的附加阴影所指出被放置在低电源状态。

更加全面的关于“主 CPU/OS 低电源”状态 304 和“非主 CPU/OS 较低电源”状态 305 的讨论将在以下立即被提供。

参照图 3 和图 4b，“主 CPU/OS 低电源”状态 304 相应于一种状态，其中该主 CPU 是加电源的并且能够执行软件；但是，整个的电源消耗与“正常开机”状态 301 比较是减少的。因为该主 CPU 能够基于该主操作系统（OS）执行软件，状态 304 被看作是“基于主 CPU/OS”的。电源可以通过以下立即描述的一种或多种技术被减小。

1) 除了 CPU 之外的具有智能/能力动态地调整它们自己的电源消耗的计算系统部件被取消它们动态调整它们自己电源消耗的权力；并且相反被强制地进入它们的最低电源状态。例如，在一个 ACPI 适应系统的情况下，在计算系统中的各种部件（包括显示器、图形控制器、各种 I/O 设备（如硬盘驱动器），等）根据 D0 到 D3 多种状态被给以调整它们自己的电源消耗的权力（例如，通过它们的设备驱动器基于注意到的用途）；这里，D0 状态是最高电源操作状态和 D3 状态是最低电源非-操作状态（D1 和 D2 也是非-操作状态，带有递增的较低电源）。在一个实施例中，该“基于主 CPU/OS 低电源”状态 304 故意地

配置特定的部件永久地保持在 D3 状态之中，只要该系统仍驻留在该“基于主 CPU/OS 低电源”状态 304 之中。非 ACPI 系统可以被类似地配置。在图 4b 中，被强制地进入它们的最低电源状态的部件是带阴影的；因此，在图 4b 的例子中，图形控制器 405、显示器 406 和 I/O 部件 408₂ 到 408_N 被强制进入它们的最低电源状态。在替换的实施例中，可以设计特定的部件使得当系统处于“基于非主 CPU/OS 低电源状态 304 中时它们的电源供给电压被减少或移去。

2) 如果 CPU 有能力动态地在多种不同的电源消耗状态之间调整它的电源消耗（例如，通过动态地改变它的内部电压级别和/或时钟频率），CPU 被强制在最低电源状态中（或该最低电源状态之间）操作。例如，基于 Intel 的 SpeedStep™ Technology 的 CPU 可能具有不同的“PN”状态：P₀ 到 P₄；其中 P₀ 是最高电源状态和 P₄ 是最低电源状态。基于 SpeedStep™ Technology 的 CPU 通过减少它的电压和频率两者来减小电源以一种适度的性能上的递减达到动态地递减电源。在一个采用基于 SpeedStep™ Technology 的 CPU 的实施例中，当计算系统是处于“非主 CPU/OS 低电源”状态 304 中时，该 CPU 牵制操作在 P₄ 状态（虽然某些应用策略可以允许特别的例外，用于进入到下一个最低电源状态 P₃）。注意其他的 CPU 可以存在，它通过减小电压或频率的某个或两者以减小电源，但不是基于 SpeedStep™ Technology 的 CPU。这样，当前被描述的技术可以用基于 SpeedStep™ Technology 的 CPU 和非基于 SpeedStep™ 的 CPU 实现。在进一步的实施例中，对于内部时钟频率范围而言，能够当在最低电源状态中时被调整，该时钟频率被设置到使处理器能够正确地操作的最低时钟频率。

3) 定义在“非主 CPU/OS 低电源”状态 304 中并不被使用的应用性软件程序并且在这个状态期间暂停它们的使用。对于“基于主 CPU/OS”的低电源状态 304 和“基于非主 CPU/OS”的较低电源状态 305 的实现被认为是没有用或不需要的任何软件任务或应用程序可以被暂停以实现非常低的系统电源。例子可包括屏幕保护程序、字处理应用程序、演示/图形应用程序、和/或电子表单应用程序。而且，任何批计算工作可以在状态 504 和 505 中操作期间被暂停。

4) 在具有多个主 CPU（也就是，图 1 的 CPU 101 实际上包括多个 CPU）的计算系统中，活动工作的 CPU 的数量被减少（例如，在具有 N 个主 CPU 的系统中在“正常开机”状态 301 期间），在“主 CPU/OS 低电源”状态 304 期间仅仅一个这样的 CPU 是活动的。

参照图 3 和 4c, 该“基于非主 CPU/OS 较低电源”状态 305 相应于一个状态, 其中该主 CPU401 是动力下降, 从而它不能够执行基于计算系统的主 OS 软件。注意在图 4c 的例子中, 该高速缓存 402 该系统内存 404, 和内存 I/O 控制部件 403 的至少内存控制器部分也被强制地进入非活动、低电源状态 (因为它们很大程度上支撑主 CPU 401 的工作以执行软件)。因为该主 CPU 是非活动的, 状态 305 是“基于非主 CPU/OS”的。而且, 至少因为该 CPU401 已经被构成非活动, 当与状态 301 和 304 比较时状态 305 是“较低电源”的。这里, 状态 305 可以看作是一个“基于非主 CPU/OS 较低电源”状态。然而, 重要的是在状态 305 期间被强制非活动的部件的精确的组合可能随着实施例而不同 (例如, 作为一个例子, 一个系统可能被设计成在“基于非主 CPU/OS 较低电源”状态 305 期间保持系统内存 404 活动, 从而该系统内存 404 能够在该状态 305 中被使用)。

示例性实现: 完整的无线电话系统

状态 301、304 和 305 的结合 (例如, 按照图 4a 到 4c 评述的计算系统电源的轮廓) 允许一个计算系统当在各种减少电源消耗的阶段中时被特殊地裁剪以减少特定的任务—并且由此结果, 产生较好的效率。一个例子帮助演示这种方法的潜力。图 5 表示示例性的“完整的”—但是有能量效率的—无线电话系统, 它能够以如同在图 3 和图 4a 到 4c 中勾画出的计算系统来实现。这里, 一个完整的无线电话系统是一种系统: 1) 提供基本的无线电话功能 (也就是一个 Plain Old Telephone Service (POTS) 接口和在一个无线电话和该 POTS 接口之间的链接); 2) 一个记录呼叫者消息的应答机器, 应该该无线电话响应于该呼叫者的电话停留于非应答情况; 和 3) 一个 Net Meeting (网络电话) 引擎, 它响应于 (正与呼叫者关联的、网络会议对它适合的) 呼叫者的 ID 建立跨越因特网的交换。

上述指出的完整的无线电话系统的一个基本的实现 (如在图 5 中所见) 将执行: 1) 基本的无线电话功能, 由来自于“基于主 CPU/OS 较低电源”状态 505 之中; 2) 应答机器功能, 由来自于“基于主 CPU/OS 低电源”状态 504 之中; 3) Net Meeting 引擎, 由来自于“正常开机”状态 501 之中。通过在该“基于非主 CPU/OS 较低电源”状态 505 中实现该基本无线电话功能, 该计算系统能够容易地如由它的用途所指示而转换它自身, 在一个“仅仅”基本无线电话功能和

一个完全许诺的计算系统之间返回或前往。参照图 4a 到 4c 作为基础的计算系统，注意：1) Net Meeting 引擎用图 4a 完整的计算系统实现；2) 应答机器用图 4b 的较低电源基于主 CPU 系统实现；和 3) 基本无线电话功能用 I/O 单元 408 实现。

注意，以上所述功能的实现是与它们相应的功能的/处理的能力和电源消耗的要求想一致的。也就是，基本无线电话功能能够容易地从少量简单的部件被构建，并且因此：1) 能够被容易地集成到单一的 I/O 单元中(例如 I/O 单元 408₁)；和 2) 将消耗少量电源(相对于整个计算系统而言)。这样，基本无线电话是一个被设计成为 I/O 单元 408₁ 的理想的功能，作为该计算系统的“基于非主 CPU/OS 较低电源”状态 505 功能之一。

比较而言，应答机器是更复杂功能，它要求存储系统用于两方面：1) 用于对呼叫者(其呼叫没有被应答)播放的记录的消息；和，2) 呼叫者的被记录的消息(如果有)。这样，尽管应答机器能够被集成到一个 I/O 单元，它更可能经济地使用计算系统的主存储器 404 的存储资源用于存储记录消息。而且，主 CPU401 和主 OS 能够被用来执行应用软件，他管理该被记录的消息回放的处理(对呼叫没有被应答的呼叫者和想要听到呼叫者的消息的计算系统的用户两者)。

还要注意，当用户没有有效地应答一个呼叫时，应答机器往往记录消息。这样，在大部分环境中，如果显示器 406 和图形控制器 405 没有开机时(例如，如果用户不在家应答电话呼叫，该用户也不能通过显示器 406 连接到该计算系统)，它不是不方便的。而且，假定主 CPU401 和主 OS 能够如上所述地被用于帮助应答机器的操作，注意该应答机器任务对于典型的计算系统 CPU401 决不是“高性能”任务。结果，如果该主 CPU401 被配置成具有一种降低性能/电源消耗状态(例如，被强制使用较低内部电压和/或较低时钟频率)，这些任务就能够被容易地完成。

取所有以上应答机器的特征在一起，注意该应答机器的功能将很好地适合于“基于主 CPU/OS 低电源”状态 504。也就是，参照图 4b，该显示器 406 和图形控制器 405 是非活动的(从而节省电源)；和，该主 CPU 和 OS 能够被用于在一个降低性能/电源功耗能力中，以处理消息回放和消息报告功能。结果，图 5 指出该完整的电话系统的应答机器部分用该“基于主 CPU/OS 低电源”状态 504 实现。这样，回顾至此图 5 的讨论，该电话系统的计算系统的实现是深思

熟虑地以计算系统的操作状态来安排的。这就是，该较低电源/较低性能基本无线电话系统功能是有意识地进入到“非主 CPU/OS 较低电源”状态 505；和，更复杂的应答机器功能是有意识地进入到“主 CPU/OS 低电源”状态 504。

最后，图 5 的该完整的无线电话系统的 Net Meeting 功能被特意用于当该计算系统是处于“正常开机”状态 501 中时。这里，负责通过因特网处理事务的软件可能涉及高性能任务（例如，层 4 的流控制，IP 层的头部处理，等）。而且，该因特网的连接除了关联于无线电话的 POTS 接口之外还可以通过其他的网络接口（例如，无线接口）建立。这样，该因特网事务可能涉及一个 I/O 单元的使用，而不是基本无线电话功能被集成在其中的 I/O 单元（例如，I/O 单元 408₂，如果 I/O 单元 408₂ 提供无线接口）。这样，该完整的无线电话系统的 Net Meeting 功能将很好地适合用于该计算系统，当它处于“正常开机”状态时（因为常规的计算系统的因特网通信能力能够被广泛地重-使用于该无线电话系统的 Net-Meeting 通信功能）。

因为该计算系统能够跨越图 5 的状态图中的各种状态转换，以上描述的完整的无线电话系统能够符合在“高电源”计算系统和“较低电源”基本无线电话之间来回摆动的计算系统。例如，仅仅作为一个例子，如果用户使用该计算系统作为传统的计算系统使用（例如，写文档和/或写文档）该计算系统将是处于“正常开机”状态 501。如果用户然后暂停这个活动，从而临时地脱离该计算系统（例如，通过做“某些事”，它不与计算机相关），该计算系统可自动地下落到它的最低电源活动状态 505（“基于非 CPU/OS 较低电源状态”）从而作为一个基本无线电话系统做的那样运转和消耗电源。注意在很多场合下用户可能有时脱离一个计算系统数小时-这样，把该计算系统自动地下降到较低电源状态，使得该计算系统作为一种在其中它被使用的方式的功能调整它自身电源消耗。

因此该计算系统还能够被看作是传统的“高电源”计算系统和低最终应用（在此情况下是基本无线电话）之间的桥梁。当用户使用该系统为传统的计算目的时（例如，文档书写，网络冲浪等），该系统作为传统的计算系统运行；和，当该用户不把系统用做传统的计算系统时，该系统降级（在功能性和电源消耗方面）成为一种基本应用（在此情况下，是基本无线电话功能）。如同图 5 中状态图所表述，指出该计算系统有能力在各种有用的状态 501、504、505

之间前后转换；同样，该计算系统有能力在传统的计算系统和基本应用之间前后转换。

而且，继续以上提供的例子，如果该用户在临时脱离该计算机之后接收一个电话的呼叫并且不能应答该电话呼叫；那么，该计算系统能够触发一个状态转换从最低电源可操作状态 505 到中间电源可操作状态 504，从而转变自身从基本无线电话（如由状态 505 所表述）到具有应答机器的基本无线电话。在这种场合，注意该系统有能力根据整个提交自身给系统的使用，来调整它的功能能力和相应的电源消耗。

继续这个同样的例子，在记录了呼叫者的消息之后（例如，通过存储它到系统内存 404），关联于“基于主 CPU/OS 低电源”状态 504 的软件可能被写入，从而拉回该系统到较低电源状态 505（缺少用户对于传统计算系统使用的返回）以至转换计算系统返回到一个基本无线电话。这样，根据这些状态的转换，注意该计算系统不仅仅能够在传统计算系统（状态 501）和基本应用（状态 505 的基本无限电话）之间调整它的功能能力和相应的电源消耗；而且同样还能够调整它的功能能力和相应的电源消耗到一个“中等”应用（状态 504 的应答机器）。而且，以上讨论的在各种功能能力之间的转换能够被自动地根据当时呈现本身给计算系统的是什么使用而触发。

一个完整的无线电话系统的例子，它表示的一系列事件足以使得网络会议被建立，将在以下利用图 11 被更详细提供。

状态转换方法和支撑硬件

已给出以上例子描述一个工作系统，它能够在各种有用的状态 501、504 和 505 之间转换它自己（每个状态有它们自己的功能性能力和电源消耗的级别），这些状态转换被实现所按照的方式是有某些重要性。图 6a,b 到 8a,b 指示这些状态转换方面。尤其是，图 6a 和 6b 提供用于在图 3 的高电源“正常开机”状态 301 和“基于主 CPU/OS 低电源”状态 304 之间状态转换的方法。图 8a 和图 8b 提供用于在“基于主 CPU/OS 低电源状态 304 和“基于非主 CPU/OS 较低电源”状态 305 之间状态转换的方法。图 7 提供电路设计的实施例，它能够被用来帮助状态转换的过程。

图 6a 展示一种方法的实施例，它可以由计算系统执行以从高电源“正常开机”状态 301 转换到“基于主 CPU/OS 低电源”状态 304。根据图 6a 的方法，

该系统起始在“正常开机”高电源状态 601 中“执行”。在此状态中，该系统能够被使用于传统的计算目的。在某些例子中，一个事件被检测 602，它触发从高电源“正常开机”状态到“基于主 CPU/OS 低电源”状态的转换过程。该事件可能随实施例的不同和应用程序的不同而不同。

例如，作为仅仅几个可能的例子，该计算系统可能识别到在一个扩展的时间周期中并没有刺激由用户所提供（例如，用户在一个扩展的时间周期中没有使用鼠标器或按动键盘）；或者该计算系统可能识别到用户已经盖上了计算系统的盖子（如果该计算系统是一个手持设备如膝上型/笔记本计算机）；或者，具有关闭电源的屏幕/显示器（如果该计算系统是一个典型的“桌上型”系统）。注意无论该事件将导致系统进入“主 CPU/OS 低电源”状态还是现有技术的睡眠状态 303 都根据不同的条件决定，它随着实施例的不同和应用程序的不同而不同。仅仅作为一个例子，如果低电源操作状态通过在软件中设定一个标志被识别为是“活动的”（例如，如果基本无线电话系统被识别为是活动的），该系统自动地转换到较低电源操作状态 304，305 而不是一种先前技术的睡眠状态 303。

响应于检测到的事件 602，该 OS 标记它自身 603 作为是在“主 CPU/OS 低电源”状态。这里，再一次回想这个低电源的但是操作性状态的“主 CPU/OS”部件意指该主 CPU 仍然是操作的和该主 OS 仍然是操作的，从而一个或多个应用软件程序能够在该主 CPU 和 OS 上执行。这样，该 OS 标志它自身 603，从而它能够正式地识别它是在较低电源状态中。适当的软件驱动器也可能类似地标志它们自身。“基于主 CPU/OS 低电源”状态 304 然后被设置或建立 604。在这种场合，回想状态 304 可以被实现，通过：1) 剥离各种部件（例如，图形控制器和显示器）调整它们自己电源消耗的权力；和/或 2) 强制 CPU 保留在一种低性能/较低电源消耗的模式中；和/或 3) 停泊特定的应用软件程序或任务；和/或 4) 在多主 CPU 系统中减少活动主 CPU 的数量；和/或 5) 对于各种部件移除或降低电源。任何对于“主 CPU/OS 低电源”状态处于不活动的软件可以把它的环境保存，从而当返回到“正常开机”状态时它能够调回它的操作环境。一旦该“主 CPU 低电源状态”被建立，该系统在这个状态中执行 605。

系统从“基于主 CPU/OS 低电源”状态转换到“正常开机”状态（它的一个实施例在图 6b 中表示）能够很大程度上作为从“正常开机”状态到“基于主 CPU/OS 低电源”状态的逆转而实现。这就是，参照图 6b 当在该“基于主

CPU/OS 低电源”状态中执行 606 时，检测到一个事件 607 它触发向“正常开机”状态的转换。此外，触发事件 607 的精确的自然特性可能随着实施例的不同和应用程序的不同而不同。在一种应答机器的场合中，如以上图 5 所描述的那样，触发事件 607 可能是 Net Meeting 所需要而建立的用于目前接收的呼叫的识别。

响应于触发事件 607，OS 和可应用设备驱动器标志它们自身 608 作为是在“正常开机”状态之中。该“正常开机”状态然后被建立/设置 609（例如，通过批准各种部件调整它们自己的电源消耗，允许主 CPU 操作在高性能/电源消耗的模式中，重新激活“停泊”的应用软件程序和重新存储它们的环境，重新为各种部件提供它们正确的支撑电源）。一旦“正常开机”状态被建立，该系统在“正常开机”状态中执行 610。

因为该主 CPU/OS 在“基于主 CPU/OS 低电源”状态期间是加电源/保持觉醒的，该主 CPU 和主 OS 在“正常开机”状态和“基于主 CPU/OS 低电源状态”之间的状态转换过程期间没有被放入到睡眠。通过比较，该“基于非主 CPU/OS 较低电源”状态的“非主 CPU/OS”部件指出主 CPU/OS 是被放入到不活动状态。结果，转换到“基于非主 CPU/OS 较低电源”包括把该主 CPU/OS 放入到睡眠状态。通常，当该主 CPU/OS 在被放入到睡眠之后而“觉醒”时，“唤醒”处理的启动阶段是类似于当整个计算系统总体上首先被加电源时或者当该计算系统由 RESET 情况出来时的那些处理。也就是，基本 BIOS 软件必须被加载和与 OS 自身一起被执行。

然而，在基本的电源加载或 RESET 响应和从睡眠状态返回之间的基本差别是：当从睡眠状态返回时，该启动软件加载过程识别到系统是从睡眠状态返回。这种识别依次地使得先前存储的环境重新装载。比较而言，当从基本加电源或 RESET 启动时，没有这种识别或环境存在。这里，一个或多个被存储在指定的被寻找位置的专用的位在唤醒期间被使用，从而该系统能够决定是从基本加电源/RESET 启动还是从睡眠状态启动（并且决定在各种情况中系统是从那种睡眠状态被唤醒）。如果一个或多个位指出该系统是从睡眠状态返回，被保流的环境被存储使系统能够返回到它原始的环境。

这些存在的寻找位指出：某些关联于 CPU 和/或内存控制器和/或 I/O 控制器功能在主 CPU/OS 正在睡眠的时间周期期间保持加载电源。这种受限制的硬件的实施例在图 7 中被表示。图 7 的特殊电路的设计不仅仅提供从睡眠状态（由

进入到“基于非主 CPU/OS 低电源状态”而被启动的)的恢复,还与从现有技术 ACPI 适应的睡眠状态的恢复相兼容。这样,参照图 3 和 7,图 7 的电路设计能够由 ACPI 兼容系统使用,以处理:1)从传统睡眠状态 303 的转换;和,2)从“基于非主 CPU/OS 较低电源状态”305 的转换。

在各种实施例中用于从睡眠状态恢复的电路(例如图 7 的电路)是与内存控制器和/或 I/O 控制器功能相集成。在进一步的实施例中,内存控制器功能以第一半导体芯片实现,而 I/O 控制器功能以第二半导体芯片实现。在此情况下被用于从睡眠状态恢复的电路(例如图 7 的电路),可能被集成到内存控制器半导体芯片或者 I/O 控制器半导体芯片。

现在返回到图 7,一个用于指出是不是该系统从一个睡眠状态或从一个基本电源/RESET 状态唤醒系统的“寻找”位相应于位 702 (WAK_STS)。决定“WAK_STS”位 702 的状态的操作将在以下被更加详细地描述。当系统作为整体进入传统“非操作性”睡眠状态时(例如图 3 的状态 303),该“SLP_EN”和“SLP_TYP”位 712,713 被写入。这里“SLP_EN”位 712 指出系统进入传统非操作性睡眠模式而“SLP_TYP”位 713 指出那种传统非操作性睡眠状态的特定类型被进入(例如在 ACPI 基本系统中 S0 到 S4),注意图 7 特定的 SLP_TYP 实施例使用 3 个位。

当系统进入传统非操作性睡眠状态时,“SLP_EN”和“SLP_TYP”位 712,713 都被唤醒睡眠逻辑 701 使用,以在该计算系统中建立适当的电源供给电压方案。这就是,每种传统的睡眠状态的类型可能具有它自己唯一的电源供给电压的方案(例如,某些部件可能使供给移去、某些部件可能使电源供给电压降低,等)。输出 709 用来实现对于指出的传统睡眠模式的正确的电源供给方案。注意,如果特定的睡眠模式方案逻辑上禁用一个或多个部件,而不是拧转他们的电源供给电压(例如,通过关闭一个输入时钟、激活一个禁用位,等等),唤醒/睡眠逻辑 701 和输出 709 可能也被用于无能力化目的。

当系统从“基于主 CPU/OS 低电源”状态 304 转换到“基于非主 CPU/OS 较低电源”状态 305 时,“NMC/O_EN”位 701 被写入。这里,因为该“基于非主 CPU/OS 较低电源”状态可能具有它自己的唯一的电源供给电压方案(例如,将对什么特定的部件使它们的供给电源移除、减小,等等),在一个实施例中,该唤醒/睡眠逻辑 701 具有一个专门的“NMC/O_EN”输入位 710 以指出:特别对于“基于非主 CPU/OS 较低电源状态”的电源供给方案将被使用。

在一个替换实施例中，即使在“基于非主 CPU/OS 较低电源”状态中，“睡眠”用语也由“SLP_EN”和“SLP_TYP”位 712，713 标记（例如，通过使用唯一的/迄今未使用的 SLP_TYP 位的组合来表示该“基于非主 CPU/OS”状态）。这里，该“NMC/O_EN”位能够被用作为附加的信息，当置位时，告诉唤醒/睡眠逻辑 701：已经被转换到“基于非主 CPU/OS 较低电源”状态。无论如何，输出 709 被用来建立正确的电源方案。再者，注意如果“基于非主 CPU/OS 较低电源”状态 305 逻辑上禁用一个或多个部件而不是拧转它们的电源供给电压（例如，通过关闭一个输入时钟、激活一个禁用的位，等），唤醒/睡眠逻辑 701 和输出 709 也可能被用于无能力化目的。

到多输入 OR 门 703 的输入位 704，714 是唤醒事件位。也就是，足以使得主 CPU/OS 从传统睡眠状态或“基于非主 CPU/OS 较低电源”状态被唤醒的事件刚到达，这些输入位 710，714 的至少一个被激活。这就使得网 708 成为活动的；它依次使得 WAK_STS 位 702 成为活动的。响应于 WAK_STS 位 702 活动，该主 CPU/OS 识别它将被从一个睡眠状态唤醒；并且，然后观察位 704，714 以进一步识别为什么该系统被唤醒。而且，依赖于实现，该主 CPU/OS 能够通过读取“NMC/O_EN”位 710 的状态或者“NMC/O_STS”位 714 的状态识别它是从“基于非主 CPU/OS 较低电源”状态被唤醒。

因为 NMC/O_EN 位 710 被置位活动以使系统进入到“基于非主 CPU/OS 较低电源”状态，在一个实施例中，位 710 能够在唤醒期间被读取以识别该系统是从“基于非主 CPU/OS 较低电源”状态被唤醒。注意在这种情况下，位 710 是一个读/写位，因为它能够被写入（为了进入“基于非主 CPU/OS 较低电源”状态）也能够被读取（为了从该“基于非主 CPU/OS 较低电源”状态转换）。在这种特定的场合，该 NMC/O_STS 位 714 被简单地用于通知图 7 的电路该系统从“基于非主 CPU/OS 较低电源”状态被移除（也就是，一个唤醒事件已发生）。

在一个替换的实施例中，该 SLP_TYP 位 713 被用于指出该系统正进入“基于非主 CPU/OS 较低电源”状态（例如，通过唯一的/迄今未使用的 SLP_TYP 位设置的组合），为了识别该系统是从“基于非主 CPU/OS 较低电源”状态被唤醒，这些同样的 SLP_TYP 位被读出。在另一个替换的实施例中，该系统被配置以注意 NMC/O_STS 位 714 来识别该系统是不是从“基于非主 CPU/OS 较低电源”状态被唤醒（也就是，如果位 714 在唤醒上是活动的；那么，该系统

从“基于非主 CPU/OS 较低电源”状态被唤醒)。位 704 是“现有技术” ACPI 位，它相应于传统的唤醒事件（例如，同 LID_STS 位相比，打开的膝上型/笔记本型计算机的原先关闭的盖子）。

图 8a 和 8b 分别展示足以把计算系统从“基于主 CPU/OS 低电源状态” 304 转换到“基于非主 CPU/OS 较低电源”状态 305 的方法（图 8a）；和，从“基于非主 CPU/OS 较低电源”状态 305 转换到“基于主 CPU/OS 低电源状态” 304（图 8b）。参照图 8a，该系统起始是在“基于主 CPU/OS 低电源状态” 801 中执行。在某些要点上，事件 802 被探测到，足以触发系统到“基于非主 CPU/OS 较低电源”状态 305（例如，图 5 的应答机器功能 504 完成了它的未应答呼叫者的呼叫的记录，并且该用户并没有返回到该计算系统）。

结果，该主 CPU 和 OS 被放入睡眠 805。这包括：1) 准备该 OS 和驱动器用于转换和环境的存储 803；和，2) 记录该主 CPU/OS 正被放入到睡眠，因为它进入该“基于非主 CPU/OS 较低电源”状态（例如，通过设定图 7 的 NMC/O_EN 位 710 和/或“SLP_EN”和“SLP_TYP”位 712, 713）并且设置 804 “基于非主 CPU/OS 较低电源”电源消耗状态（例如，通过对 CPU、内存控制器、系统存储器等降低供电电源水平；例如，如同由唤醒/睡眠逻辑 701 所提供）。在这些被完成之后，该系统从“基于非主 CPU/OS 较低电源”状态执行 806。

图 8b 提供更一般的该计算系统可能遵循的唤醒序列，如同它从传统的睡眠状态 303 或从“基于非主 CPU/OS 较低电源”状态 305 唤醒。根据图 8b 的过程，唤醒事件 807 触发主 CPU 从睡眠状态退出并且 BIOS 软件被装载 808。在从传统睡眠状态转换的情况下，唤醒事件 807 可能是一个指示该用户已经返回到使用该系统（例如，与图 7 的 LID_SIS 输入相比较通过打开关闭的笔记本/膝上型电脑的盖子）。在从“基于非主 CPU/OS 较低电源”状态转换的情况下，该唤醒事件 807 可能由需要使用一个功能（它不能在较低电源状态中操作）而引起（例如，依照未应答的呼叫的一个应答机器）。

只要启动恰当的软件响应于被加载的 BIOS，系统的控制被交出给该主 OS，它相应于从传统的睡眠状态 303 或从“基于非 CPU/OS 较低电源”状态 305 的转换，决定 809 是不是该唤醒事件 807。这里，依赖于实现，该主 OS 能够参照图 7 的位 712, 710 和/或 714 作出这个决定 809。如果该系统从传统的睡眠状态 303 转换，先前技术的唤醒序列可能遵从 810。如果主 OS 决定系统从“基于非主 CPU/OS 较低电源”状态转换，主 OS 将试图理解 810 为什么该基于主

CPU/OS 的子系统生成该触发事件。

这里,如果该非主 CPU/OS 较低电源子系统包括一个执行软件的处理器(如下用图 9 更详细地解释的那样),该主 CPU/OS 能够发送消息到该主 CPU/OS 较低电源子系统以决定为什么该触发事件 807 被生成。如果该非主 CPU/OS 子系统没有执行软件,该主 OS 能够观察例如附加的硬件位,他由非主 CPU/OS 设置以通知主 OS 关于该触发事件 807 的种类。

注意,图 7 的位 710 能够被取消活动,使得图 7 的唤醒/睡眠逻辑 701 正确地给硬件加电源成为“正常开机”状态(例如,如在图 4a 中表示),如果该转换是来自传统的睡眠状态 303。同样地,依赖于实现,位 710 和/或位 712 能够被取消活动,使得唤醒/睡眠逻辑 701 正确地给硬件加电源成为“基于主 CPU/OS 低电源”状态(例如,如在图 4b 中表示),如果该转换是来自“基于非主 CPU/OS 低电源”状态 305。

基于非主 CPU/OS 较低电源状态的硬件和软件设计

图 9a 和 9b 现在被用于支撑对于“基于非主 CPU/OS 较低电源”状态的可能的硬件和软件设计的更详细的讨论。这里,现在的讨论能够被看作是更详细的讨论细节,它原始地由图 4c 呈现。简要地往回参照图 4c,用于实现“基于非主 CPU/OS 较低电源”状态的硬件的简单化了的版本的重新调用被提供;这里,更具体地,单一的 I/O 单元 408₁ 被激活以实现该非主 CPU/OS 状态 305。

通过比较,更详细描述非主 CPU/OS 状态 305 的硬件实现目前被设想,它能够被看作是具有识别能力的软件和 I/O 的较低电源/较低性能的计算系统。图 9a 展示一个例子。在图 9a 的描述中,一个具有基于主 CPU/OS 系统的计算系统被展示(它包括主 CPU901,高速缓存 902,系统内存 904,图形控制器 905,主显示器 906,等)。当在该“基于非主 CPU/OS 较低电源”状态时,基于主 CPU/OS 系统的很多主要的部件被取消活动。类似于图 4a 到 4c 的方案,在“基于非主 CPU/OS 较低电源”状态期间被取消活动的部件被画成在图 9a 中的阴影区域。

这里,根据图 9a 的实施例,用于实现“基于非主 CPU/OS 较低电源”状态的计算系统包括它自己的特色:1) 控制器 917(作为和主 CPU 比较,它可能用一个较低性能/较低电源处理器、微控制器、逻辑状态机等实现);2) 主存储器 918(它可能以低速随机访问存储器(RAM)和/或与主系统存储器 904 相

比较较少密度的 RAM 来实现); 3) 用户接口 925 (它可能包括一个显示器 919, 键盘/按钮 920 和 LED924); 4) 系统总线 923; 5) I/O 单元 922 (仅仅作为一个例子, 它可能用存储资源如基于 FLASH 的存储器或基于聚合物的存储器而实现); 和, 6) FLASH 存储器 921。

除了以上突出的特色特点之外, 注意用于实现该“基于非主 CPU/OS 较低电源”状态的计算系统还可以与主 CPU/OS 计算系统一起共享各种 I/O 单元。在图 9a 的实施例中, 该共享的 I/O 单元包括: 1) 一个 MODEM 单元 909; 2) 一个无线局域网 (WLAN) 单元 911; 和, 3) 一个无线广域网 (WWAN) 单元 913。这里, 这些“被共享”的 I/O 单元 909、911、913 在“基于非主 CPU/OS 较低电源”状态中是活动的。其它被共享的 I/O 接口单元也可能同样 (例如, 蓝牙 (Bluetooth))。在各种实施例中, “被共享”的 I/O 单元在一对计算系统中操作 (主 CPU/OS 和非主 CPU/OS), 这是当处于“基于非主 CPU/OS 较低电源”状态中时, 通过从该主 CPU/OS 系统取得命令进行; 和当处于“基于非主 CPU/OS 较低电源”状态中时, 通过从该非主 CPU/OS 系统取得命令进行。还可能共享的 I/O 单元可被插入到该主系统总线 907 中 (例如, 如由通信接口 926 表示的 I/O 单元 908_N)。

在一个实施例中, 接口 910、912、914 和 916 (它们在“基于非主 CPU/OS 较低电源”状态期间是不活动的) 分别地相应于: 1) 对于接口 910, 一个串行端口接口 (在此情况中 MODEM909 可进一步相应于 V.90 MODEM 909); 2) 对于接口 912, 一个 USB 接口用于一个基于 IEEE 802.11 的网络接口卡; 3) 对于接口 914, 一个串行接口用于通用分组无线电服务 (GPRS) 无线调制解调器; 和, 4) 对于接口 916, 一个 ATA-100 接口用于 IDE 硬盘驱动器 (HDD)。在进一步的实施例中, 一个从内存的 I/O 控制器部分和 I/O 控制功能 903 (为了简单化在图 9a 中未示出) 发源的通用串行总线 (USB) 接口在“基于非主 CPU/OS 较低电源”状态中被取消活动, 并且包括具有一个蓝牙 I/O 单元。这里, 蓝牙接口单元还可以被共享, 因此在“基于非主 CPU/OS 较低电源”状态期间是活动的。

在另一个实施例中系统总线 923 能够是同样的系统总线 907。在这个情况中, 在处于“正常开机”状态 301 中时, 该主 CPU901 除了访问控制器 917 以外还能够访问设备 908₁ 到 908_N。然而, 当该系统是处于“非主 CPU/OS 较低电源”状态 304 时, 该控制器 917 然后能够通过系统总线 923/907 (在这个例子

中，它们是同样的总线，但为了清楚未在图 9a 中表示) 访问设备 908₁ 到 908_N。注意，在这个实施例中，设备 908₁ 到 908_N 在该“非主 CPU/OS 较低电源”状态 304 中为了使该控制器 917 能访问它们将保持活动（非共享）。

注意，该非主 CPU/OS 系统可包括它自己的特色用户接口 925。图 9a 的实施例指出特色用户接口包括一个 LED 924（它的状态由控制器 917 控制）、一个显示器 919、和一个键盘/按钮 920。该用户接口 925 的机械布置/布局可以加入到它的“基于非主 CPU/OS 较低电源”状态方面的特色。特别在移动计算应用（例如，膝上型/笔记本计算机）中，用户接口 925 可以被定位/布置使得用户能够观察显示器 919 和 LED 924；和/或当主显示器 906 的盖子关闭而覆盖了主键盘时，使用键盘/按钮 920。图 9b 展示一对膝上型/笔记本计算系统，每个都具有用户接口，当该计算系统的“盖子”是关闭时，它能够被访问。

在这个关闭盖子用户接口上能够被访问的数据的例子是日历，联系和制作信息；通常被称为个人信息管理（PIM）数据；然而，它并不限制于这种类型的数据并且能够包括任何信息，它对于一个使用笔记本电脑处于“关闭盖子”状态的最终用户可能是重要的（例如，对于旅行的销售员的当前的销售数据）。另外，整个计算系统可能允许在笔记本电脑中通过关闭盖子的用户接口而用于功能的控制。这种例子可以是存储在计算机上的 MP3 音乐文件通过无线头戴耳机的播放。在这个例子中用户可以通过关闭盖子的用户接口控制音乐（歌曲选择、音量、和其它属性）的播放。

返回参照图 9a，注意该非主 CPU/OS 计算系统可以用控制器 917 实现，这可能依次作为微处理器或微-控制器来实现。结果，实施例被预想其中非主 CPU/OS 计算系统执行它自己的软件例程。图 10 展示图表演示非主 CPU/OS 计算系统（图 10 的右手边）的软件如何能与由主 CPU 和 OS 执行的软件协作。也就是，回想在“活动开”状态和“基于主 CPU/OS 较低电源”状态两种状态期间，该非主 CPU/OS 计算系统可保持电源开和活动，这就可能，一个“双系统”可以被实现，其中软件来自两个系统（主 CPU/OS 和非主 CPU/OS），在“活动开”状态或“基于主 CPU/OS 较低电源”状态两种状态期间，彼此操作作为一个协作整体。图 10 试图演示这个关系。

图 10 能够被看作（尽管不应该被解释是单独限制于它）是一个系统的实施例，它利用一个关闭盖子用户接口（如在图 9b 中展示的那些）使得当该膝上型/笔记本电脑的盖子是关闭的时，访问有关的最终用户数据或控制有用的

最终用户功能。图 10 表示操作功能如何在图 3 中描述的不同系统状态（“正常开机”状态 301、“基于主 CPU/OS”状态 304 和“基于非主 CPU/OS”状态 305）之间被分配。

图 10 的右边边展示该非主 CPU/OS 计算系统的软件部件。这些包括非主操作系统（也就是，非主 OS）部件如：1）一个应用程序程序员接口（API）1001；2）一个管理功能 1002（它能够包括事件管理和功能管理例程两者）；3）数据存储管理功能 1003（以控制该非主 CPU/OS 计算系统的特色数据存储资源（例如图 9a 中 FLASH 或聚合物存储器 922）的使用；和，4）一个用户接口管理功能 1004（以控制该非主 CPU/OS 计算系统的特色用户接口（例如，图 9a 的用户接口 925）的使用）。

应用软件 1005、1006 也可以驻留在非主 CPU/OS 计算系统上。应用软件能够典型地分解成两种类型：1）数据储存 1005（它指向已存储数据的使用/管理）；和，2）功能 1006（它指向由基层控制器 917 实行的有用的功能）。

作为具有典型的软件环境，该非主 CPU/OS 计算系统应用程序 1005，1006 与非主 CPU/OS 计算系统通过一个 API 1001 接口。

当在“基于非主 CPU/OS 较低电源”状态中时，该非主 CPU/OS 计算系统（包括软件部件 1001 到 1006）独立地操作。而且，因为主 CPU/OS 在“基于非主 CPU/OS 较低电源”状态期间是非活动的，运行在主 CPU/OS 上的软件部件 1007 到 1012 同样是非活动的。然而，当整个系统是处于“正常活动”状态或“基于主 CPU/OS 低电源状态”时，各种在主 CPU/OS 上运行的软件部件都是活动的；并且，因为该非主 CPU/OS 计算系统在任何这些状态期间是保持活动的，来自两个系统的软件可能工作在一起作为一个协作整体。

例如，通过认识到在非主 CPU/OS 计算系统控制之下的特定资源，该主 CPU/OS 边的软件可以被配置来“使用”这些资源。例如，在主 CPU/OS 计算系统上的软件例程可以被配置来利用数据储存或内存资源，这是对非主 CPU/OS 计算系统的特色（例如像图 9a 的单元 918、921、和 922）。作为另一个例子，在主 CPU/OS 计算系统上的软件例程可以被配置来影响关联于用户接口的各种资源的状态，这也是对于非主 CPU/OS 计算系统的特色。例如，如以下将更详细解释，在“基于主 CPU/OS 低电源”状态中实现的一台无线电话应答机器（如关于图 5 所论述）可以要求重复地闪动“开和关”一个 LED（例如图 9a 的 LED 924），这是关联于基于非主 CPU/OS 用户接口（例如，通知用户：

消息已经被记录供用户听取)。这样的实施例的描述将在以下相关于图 11 而更详细地解释。

为了对于该主 CPU/OS 软件 1007 到 1012 能与非主 CPU/OS 软件 1001 到 1006 “一起工作”，这种软件可能发送信息到非主 CPU/OS 控制器 917 以要求特定的动作或能够为储存器 1003 传递数据对象。该应用软件可再次被分解成数据应用程序 1009，1010 和功能应用程序 1013，1011。然而，这些应用程序 1009，1013 的某些可能以一种理解被预-编写，这个在非主 CPU/OS 计算系统上的资源是有效的；然而，其他的（例如，早期的“遗物”）软件应用程序 1010，1011 可以被编写而没有任何的识别或认识到这样的资源的存在。对于这些后一类型的软件应用程序，行为如同一个“粘合层”的“代理”软件 1007，1008 可被用于强制或者说成使得该遗留的应用程序能够与非主 CPU/OS 系统资源协同地操作。功能块 1002，1003 和 1004 允许用户与用户接口 925 交互作用以显示 PIM 和其他信息，或者控制某些像播放 MP3 文件那样的功能。管理程序 1002 接受数据对象，它然后被存储在储存块 1003 中。这些数据对象呈现将被显示在该用户接口上的（PIM 或其他数据）和原始在数据应用程序 1005，1010 和 1009 中的数据。应用程序 1010 和 1007 在“正常开机”状态 301 中的操作并且经该管理程序 1002 通过 API 1001 负责提供被存储在储存器 1003 中的数据对象。Legacy Data Application 1010 的例子是当前的 Outlook™ 和 Lotus Notes™，它们包含 PIM 数据，但是并不知道如何建立用户接口 1004 能够理解的数据对象或者如何移动这些对象到储存器 1003。该代理应用程序 1007 负责这个功能，并且从该 Legacy 应用程序本质上取出相应的数据，格式化它成为正确的数据对象，并且然后经 API 1001 传递这些数据对象给管理程序 1002 由储存器 1003 存储。在将来，这些类型的应用程序将插入这些输出的功能并且由数据应用程序 1009 呈现。

将在“基于主 CPU/OS”状态 304 中操作和由这个用户接口 925/1004 控制的功能包括应用程序 1011，1008 和 1013。另外，Legacy 功能呈现某些事情，这是该非-主 CPU/OS 功能未察觉的，并且因此要求一个主 CPU/OS 代理驱动器来与这些功能接口。一个这样的应用程序的例子是遗留的媒体播放器，它能够播放音乐文件。在这个场合，一个代理应用程序 1008 能够被写入以允许当在基于主 CPU/OS 状态 304 中时，用户接口 1004 控制这个应用程序。这个应用程序将允许用户控制在基于主 CPU/OS 状态 304 中有效的、存储在子系统上的

媒体歌曲的播放，并且然后通过某些音频接口输出。在将来，这些类型的应用程序将插入到这些代理功能中并由功能应用程序 1013 呈现。

在基于非主 CPU/OS 状态 305 中操作的功能驻留在该图的右方。用户接口 1004 负责对用户按钮的按动（键盘/按钮 920）起作用并且然后在显示器 919 上显示该数据对象。在数据对象中嵌入的是导航命令，这将基于哪个按钮被按动来告诉用户接口下一个将显示哪个对象。另外该管理程序将允许用户通过一个 MP3 基于非主 CPU/OS 较低电源状态的功能应用程序 1006 来控制 MP3 的回放；1006 负责从储存器 1003 取得 MP3 文件，解码该文件和通过蓝牙接口输出到一个无线头戴耳机。

数据存储应用程序 1005 的例子是一个应用程序，它连接回到企业服务器以再现新的 PIM 数据对象，它然后能够被放入到储存器 1003 用于由用户通过用户接口 1004 访问。在这种场合中，应用程序 1005 将通过在基于非主 CPU/OS 状态 305 中操作的无线通信设备（例如，WALN 911 或 WWAN 913）来访问这些新的数据对象。

图 11 展示状态转换的另一个实施例，这可能如按图 5 所描述在完整的无线电话系统的路线上发生。然而，图 11 的例子比图 5 的讨论稍许更加细致，这是指 LED 的闪动（向用户指示从一个未知电话呼叫来的信息已经被记录并等待该用户）和一次网络会议被建立两者。根据图 11 的方法，在时间周期 T1 上，该系统原始地处于“基于非主 CPU/OS 较低电源”状态 505、1101，这里除了基本的无线电话之外没有计算机的活动性/使用发生。

在时间 T2 点一次呼叫在无线电话中形成并且无人应答。结果一次转换进入到“基于主 CPU/OS 低电源状态”504、1102 使实现应答机器功能。在时间 T3 上，该应答机器功能应答该电话，播放一段消息给呼叫者，记录呼叫者的消息并使 LED 在用户接口上重复闪亮，这是该非主 CPU/OS 计算系统的特色（例如，图 9a 的 LED 924）。注意这后一个使 LED 闪动的功能相应于驻留在主 CPU/OS 上的应答机器软件之间的一个协同 workflow；并且 LED 软件/硬件是该非主 CPU/OS 计算系统的特色。

随着以上功能的完成，在时间 T4，整个计算系统转换返回到该“基于非主 CPU/OS 较低电源”状态 505、1103。在时间 T5，随着 LED 继续闪动（也就是，它没有停止闪动直到用户接听记录的消息），该电话再一次铃响。然而，这时用户应答该电话并且识别当前的呼叫是关于一次需要被建立的网络会议。用户

按动在非主 CPU/OS 计算系统用户接口上发现的“网络会议”按钮（例如，图 9a 的相关联的键盘/按钮 920）。

“网络会议”按钮的按动引起第一次转换到“基于主 CPU/OS 较低电源”状态 505、1104 和第二次转换到“正常开机”状态 501、1105。在“正常开机”状态中，该进入的电话呼叫的呼叫者 ID 和主 CPU/OS 的资源被利用来建立一次网络会议，并且在此之下履行工作。（例如，和另一个个人在因特网上通过修改一个公用的文档）。该 LED 在非主 CPU/OS 计算系统的控制下继续闪动，因为该用户仍然没有接听该记录的消息。

还应该理解因为本发明的实施例可能作为一个或多个软件程序来实现，本发明的实施例可以在一种机器可读介质之上或之内实行或实现。一种机器可读介质包括任何用于存储或传输信息的机制，按由机器可读的形式（例如，一台计算机）。例如，一种机器可读介质包括只读存储器（ROM）；随机访问存储器（RAM）；磁盘储存介质；光储存介质；快闪存储设备；电的、光的、超声的或传播信号的其它格式（例如，载波、红外信号、数字信号，等）；等等。

在上面的说明中，已经参照特定的示例性实施例描述了本发明。然而很明显：可以在那里进行各种修改和改变而没有脱离如在所附权利要求中设定的本发明广泛的精神和范围。本说明和附图相应地作为示例而非限制。

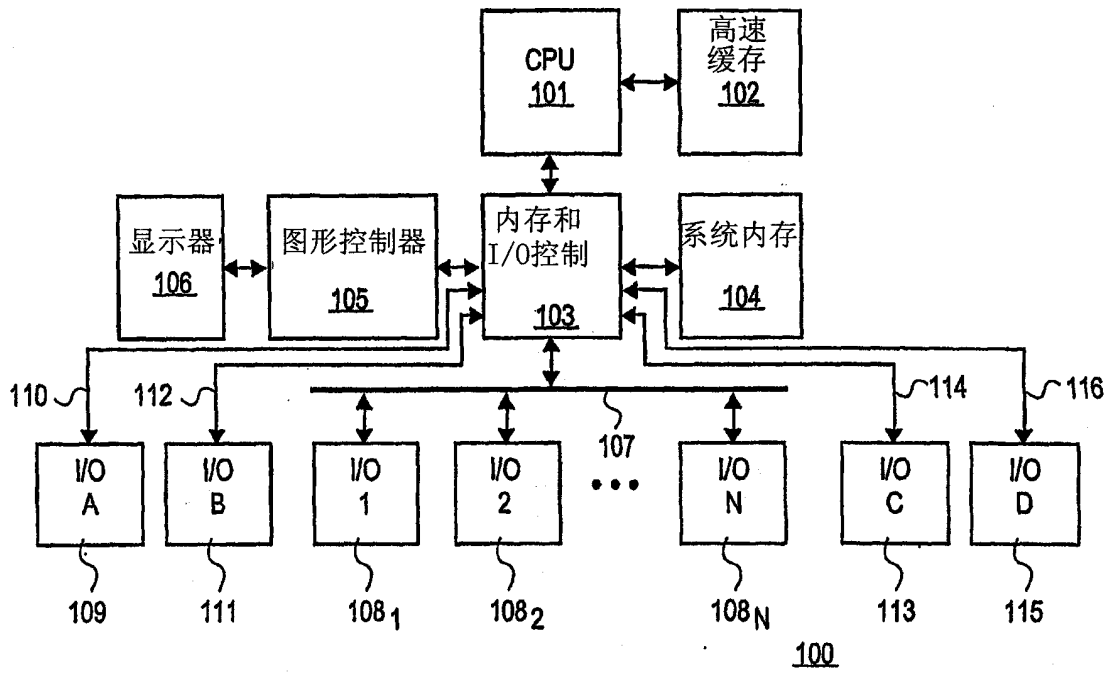


图 1



图 2

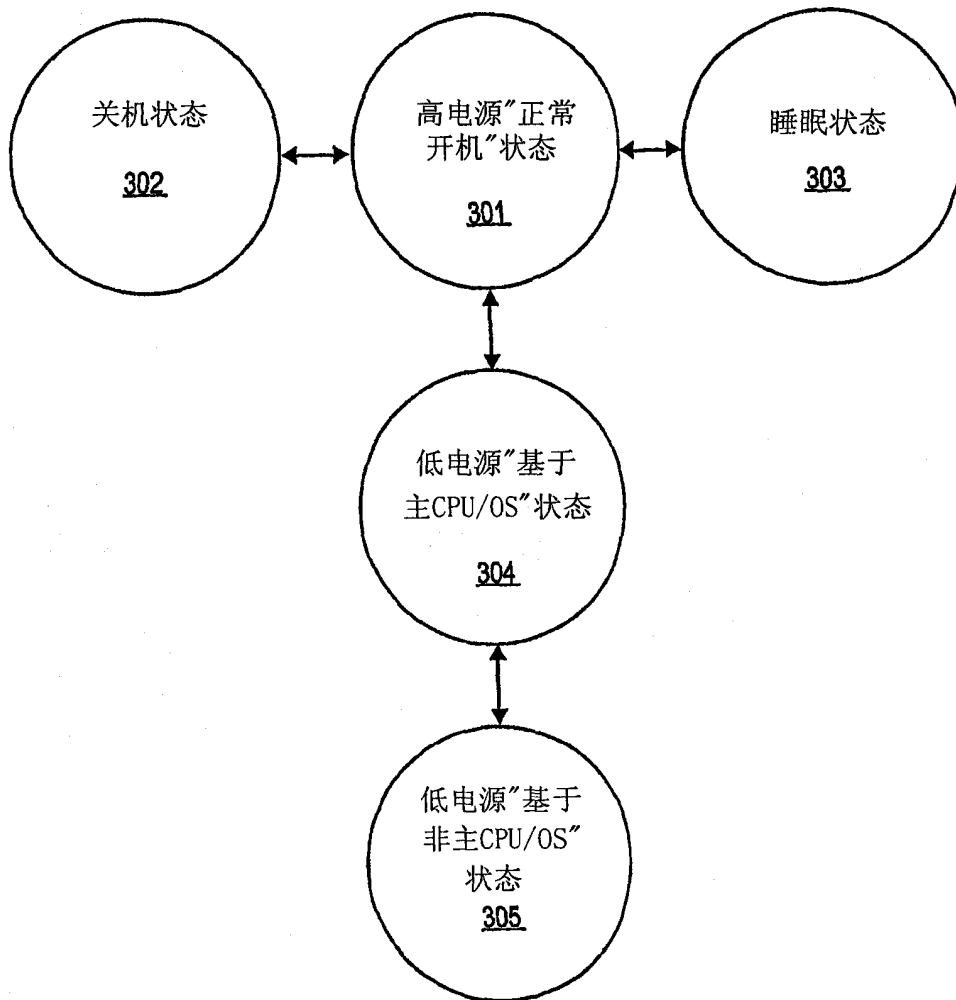


图 3

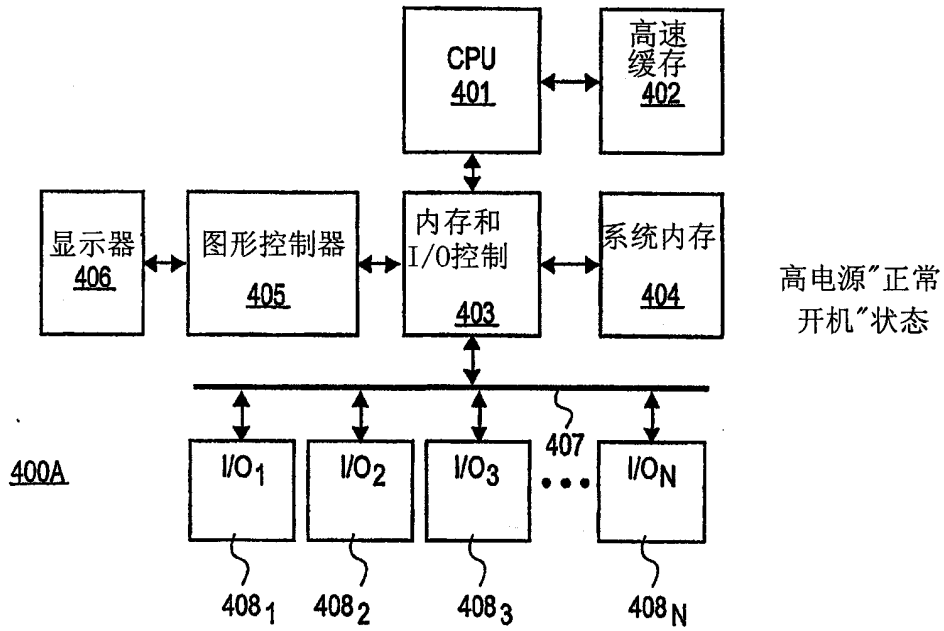


图 4A

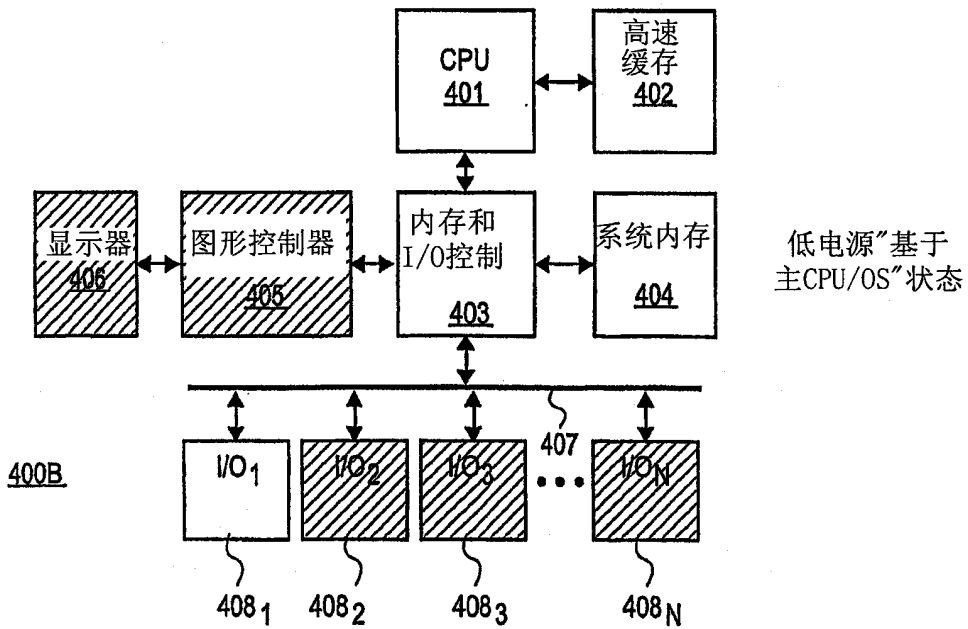


图 4B

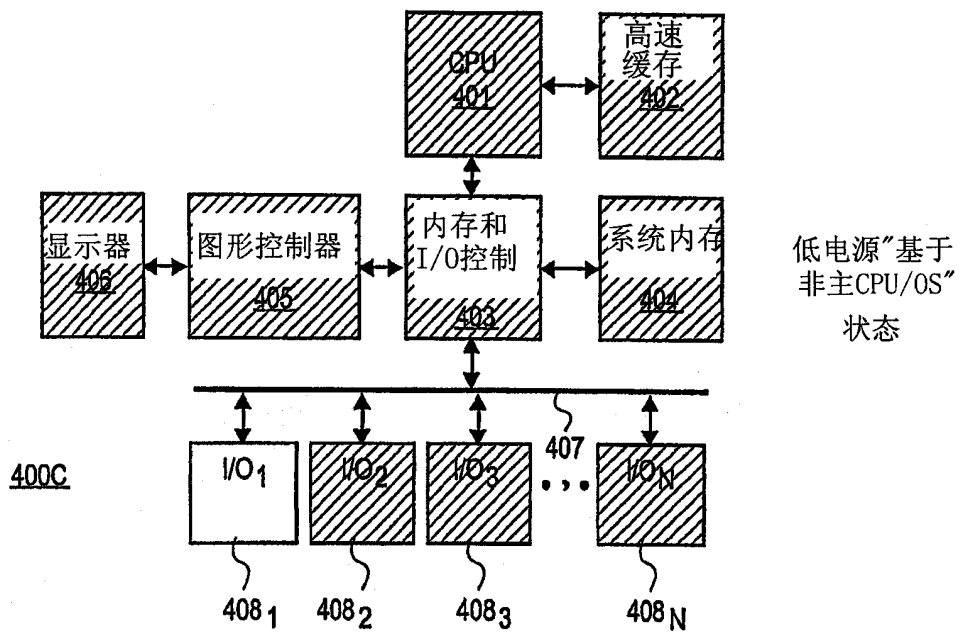


图 4C

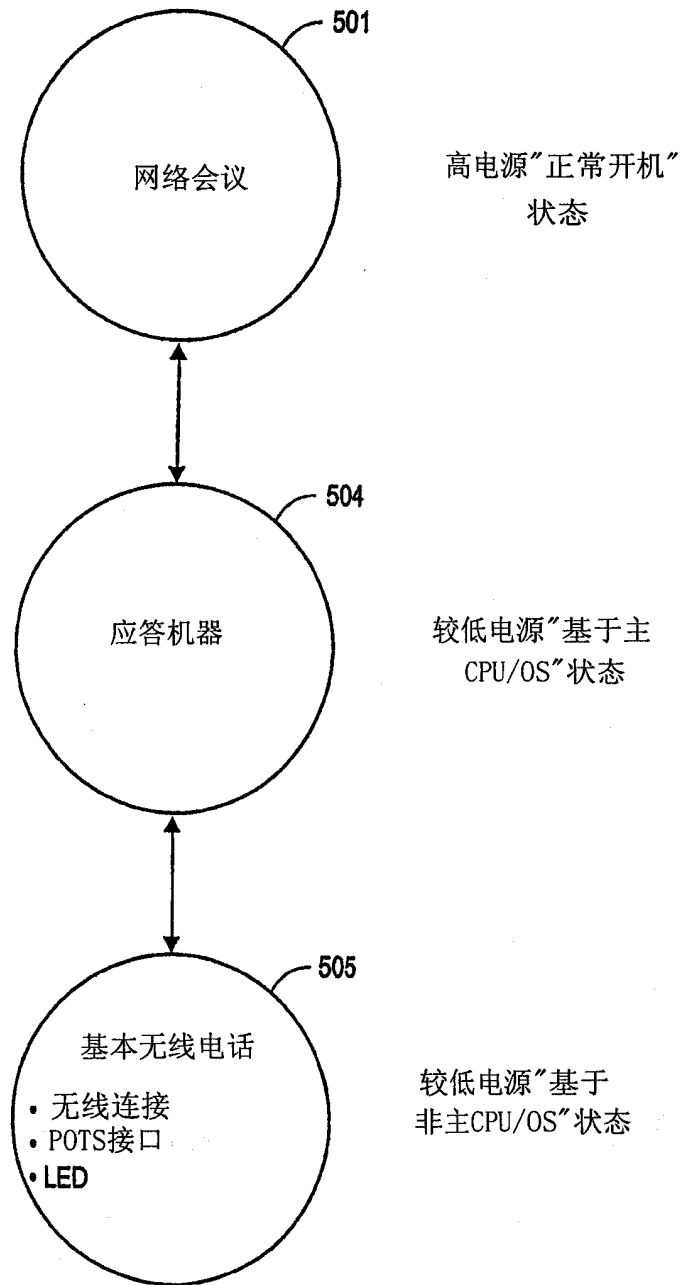


图 5

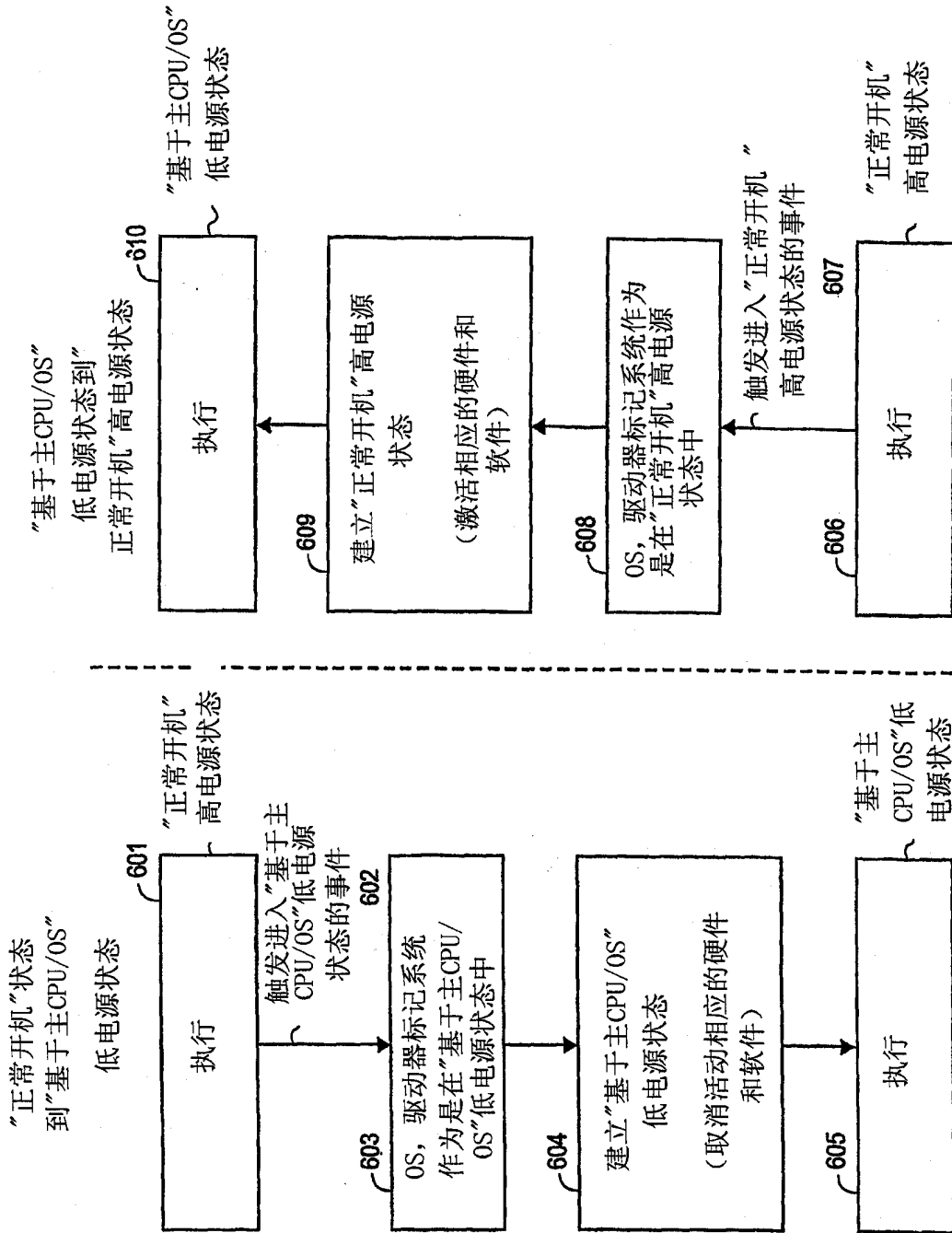


图 6B

图 6A

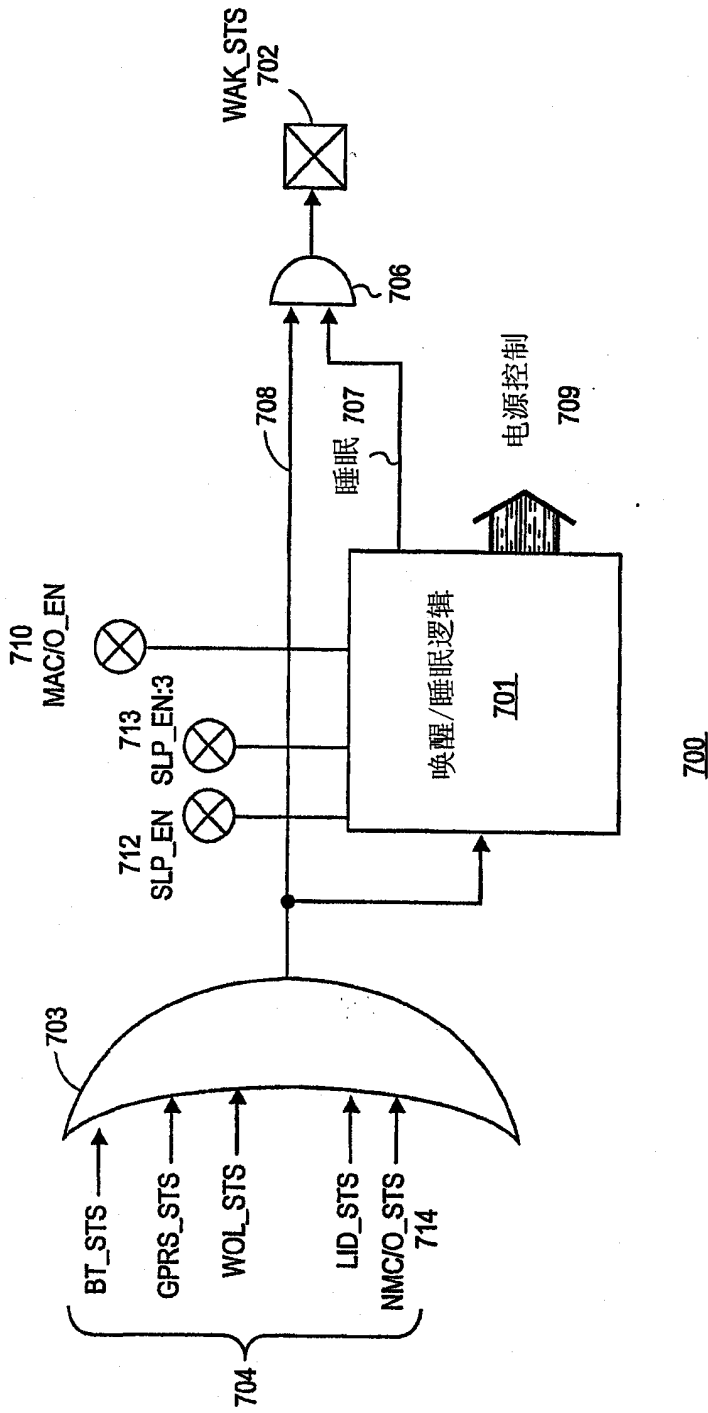


图 7

“基于主CPU/OS”低电源状态到
基于非主CPU/OS”较低电源状态

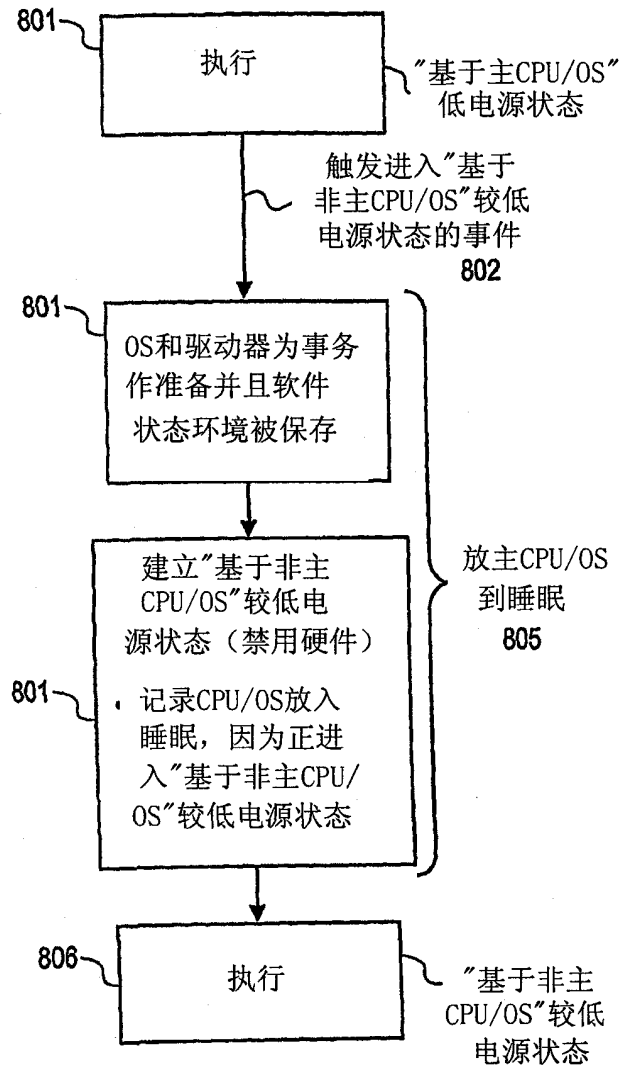


图 8A

普通的唤醒例程

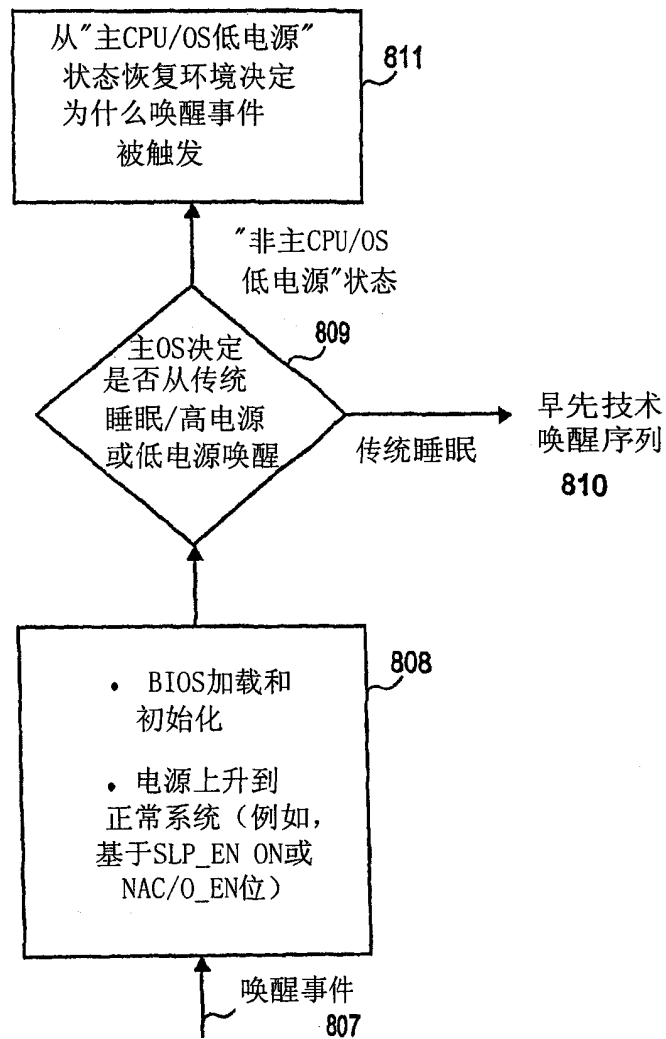


图 8B

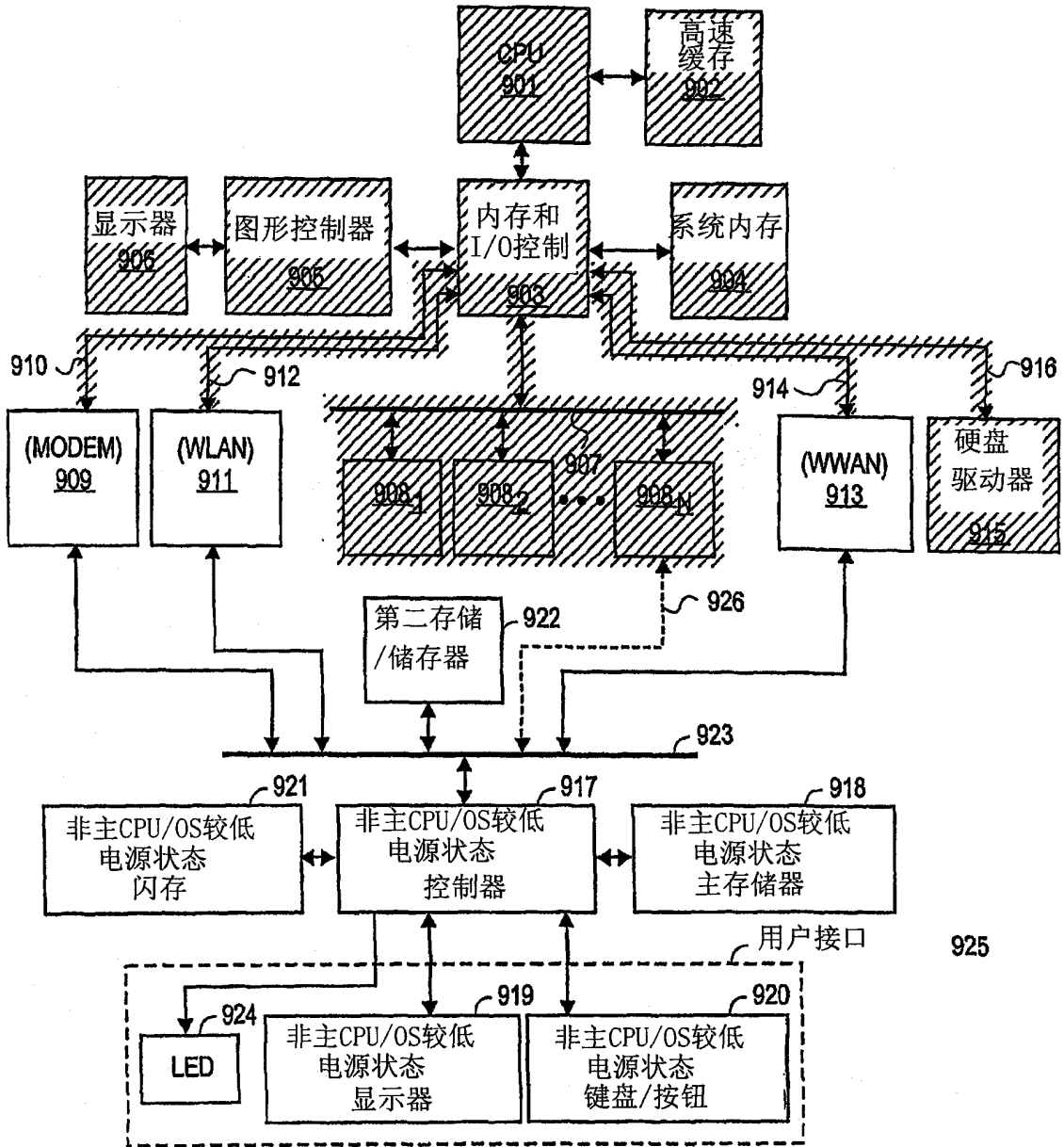


图 9A

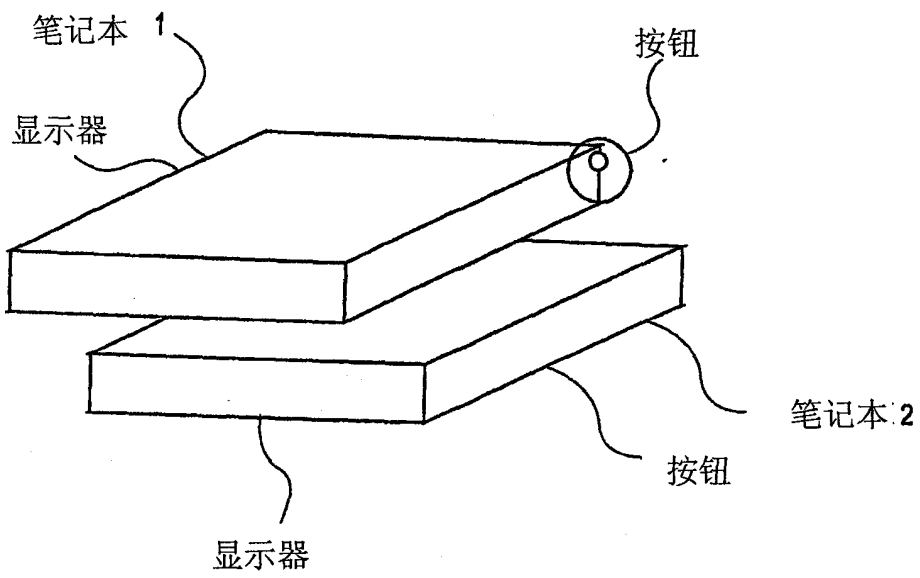


图 9B

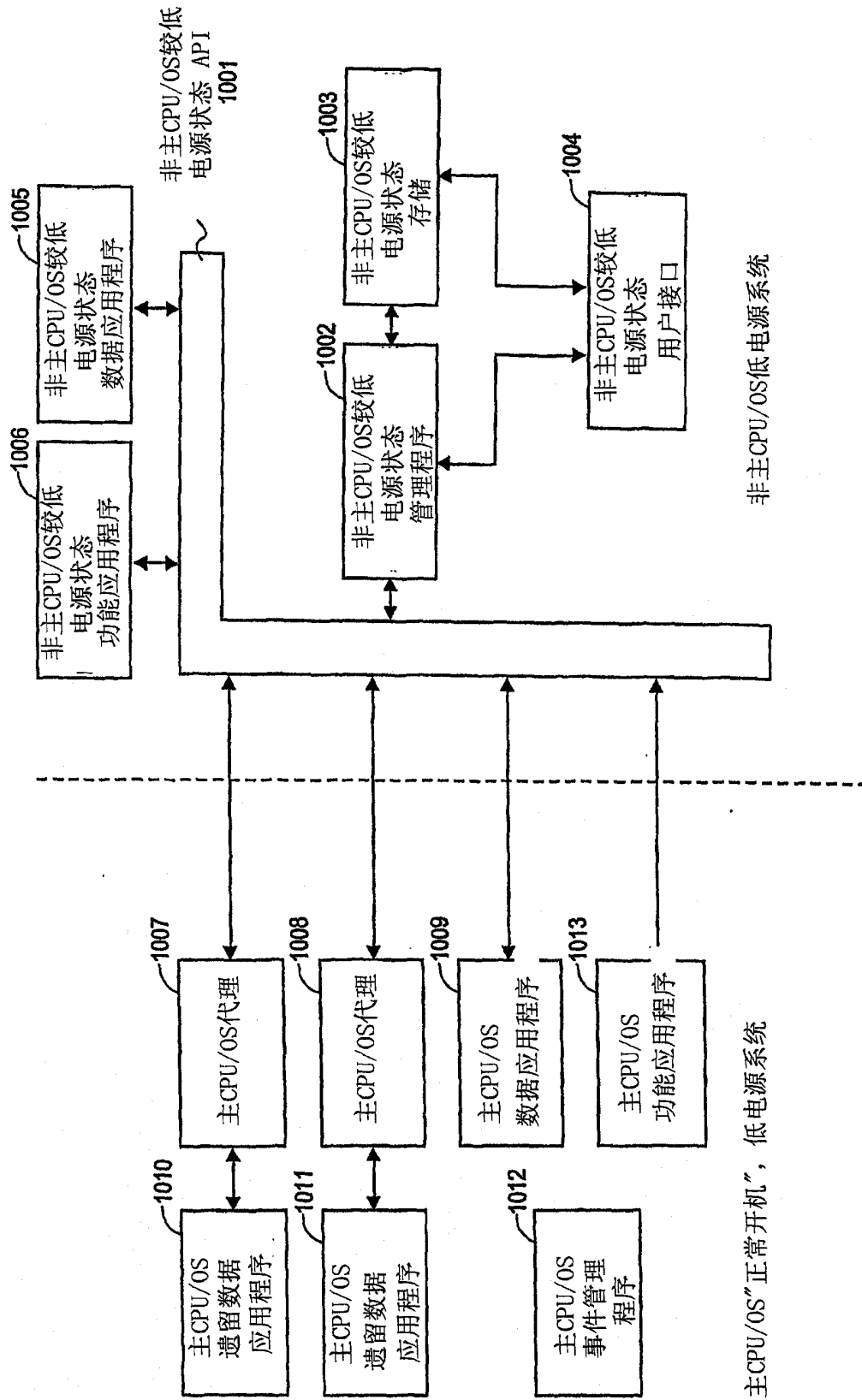


图 10

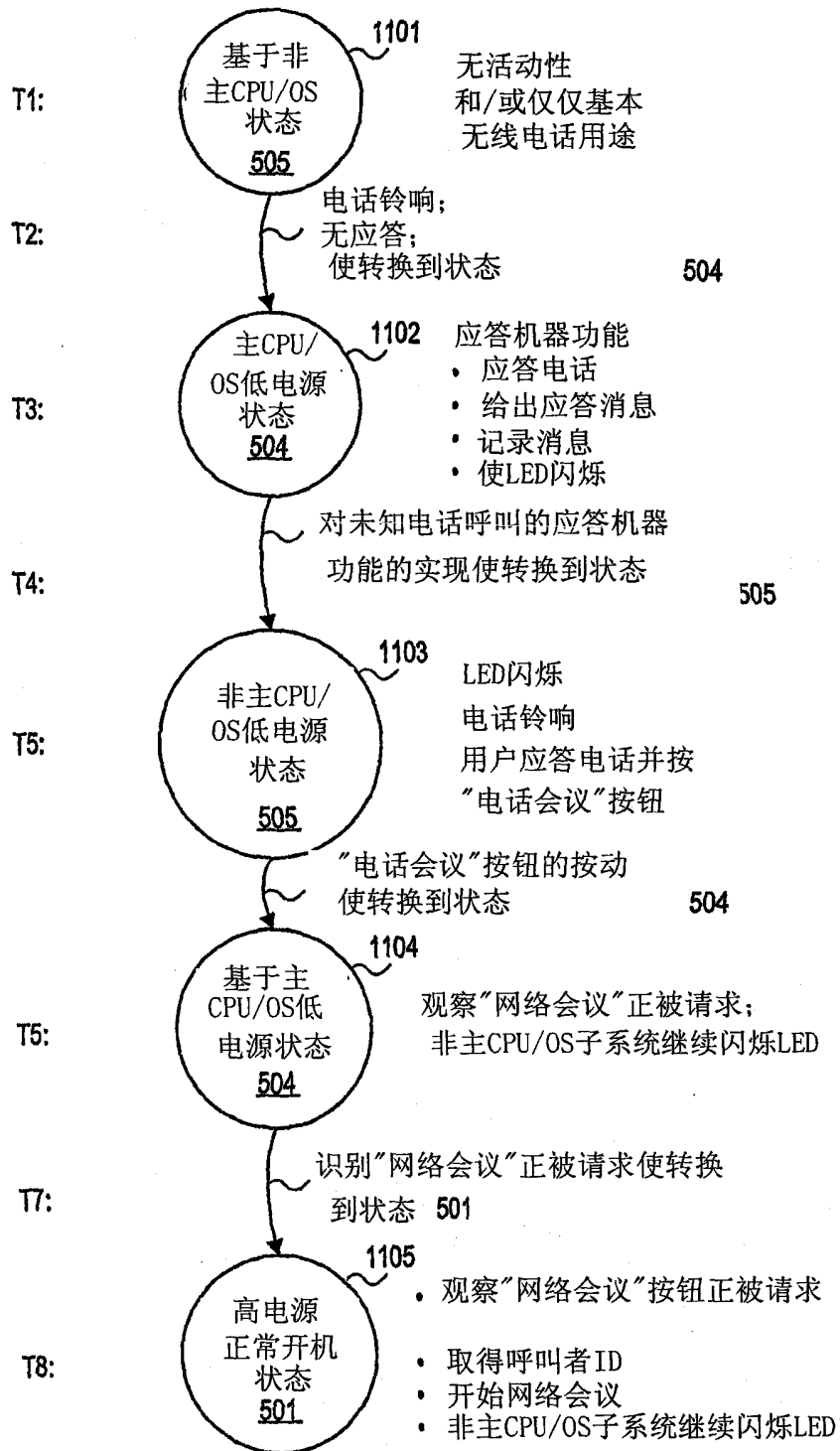


图 11