



(12) 发明专利申请

(10) 申请公布号 CN 105074663 A

(43) 申请公布日 2015. 11. 18

(21) 申请号 201480013782. 0

(51) Int. Cl.

(22) 申请日 2014. 03. 10

G06F 9/455(2006. 01)

G06F 21/53(2006. 01)

(30) 优先权数据

13/796, 442 2013. 03. 12 US

(85) PCT国际申请进入国家阶段日

2015. 09. 10

(86) PCT国际申请的申请数据

PCT/US2014/022731 2014. 03. 10

(87) PCT国际申请的公布数据

W02014/164536 EN 2014. 10. 09

(71) 申请人 高通股份有限公司

地址 美国加利福尼亚

(72) 发明人 T·曾 A·托兹尼 P·小米勒

P·帕特尔

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 张立达 王英

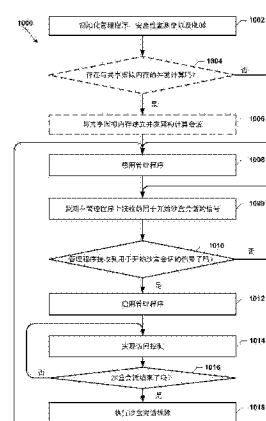
权利要求书8页 说明书23页 附图20页

(54) 发明名称

按需选择性地启用虚拟机监测器的操作的方法和装置

(57) 摘要

在各个方面,可以使用虚拟化技术来通过在沙盒会话期间选择性地启用操作在计算设备上的管理程序来改善性能和降低消耗的功率的量。在各个方面,高级操作系统可以分配内存,使得其中间物理地址等于物理地址。在管理程序被禁用时,管理程序可以暂停从中间物理地址至物理地址的第二级转换。在沙盒会话期间,管理程序可以被启用并且重新开始执行第二级转换。



1. 一种管理计算设备上的内存的方法,包括:
初始化管理程序、安全性监测器以及高级操作系统 (HLOS);
在初始化之后,禁用所述管理程序;
监测来自所述安全性监测器的用于开始沙盒会话的信号;
在所述信号被接收以开始所述沙盒会话时,启用所述管理程序;以及
当所述管理程序被启用时,实现访问控制。
2. 根据权利要求 1 所述的方法,其中,所述安全性监测器是 ARM 信任区®。
3. 根据权利要求 1 所述的方法,其中,所述管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或被启用。
4. 根据权利要求 1 所述的方法,其中,初始化所述管理程序包括配置所述 HLOS 来分配内存空间,使得所述 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。
5. 根据权利要求 4 所述的方法,其中,初始化所述管理程序还包括利用所述安全性监测器对所述管理程序的代码和数据进行认证。
6. 根据权利要求 5 所述的方法,还包括当所述管理程序被启用时,配置所述管理程序的代码和数据对数字信号处理器和被包括在所述数字信号处理器中的 CPU 中的至少一个不可访问。
7. 根据权利要求 1 所述的方法,其中,禁用所述管理程序包括:
配置所有系统内存管理单元 (SMMU) 的上下文库来绕过第二级转换;以及
针对所述 HLOS 关闭第二级转换。
8. 根据权利要求 7 所述的方法,其中,禁用所述管理程序还包括下列各项中的至少一项:
暂停限制所述 HLOS 对硬件中断的访问;
暂停限制所述 HLOS 对硬件定时器的访问;以及
暂停限制所述 HLOS 的 I/O 访问。
9. 根据权利要求 1 所述的方法,还包括:
确定所述沙盒会话是否结束;
在确定所述沙盒会话结束时,执行沙盒会话拆除过程;以及
在执行所述沙盒会话拆除过程之后,禁用所述管理程序。
10. 根据权利要求 9 所述的方法,其中,确定所述沙盒会话是否结束包括接收指示所述沙盒会话结束的另一信号。
11. 根据权利要求 9 所述的方法,其中,执行所述沙盒会话拆除过程包括:
释放用于沙盒化的组件的所有缓冲器;以及
恢复第二级转换页表,以移除所有存储残片。
12. 根据权利要求 1 所述的方法,其中,启用所述管理程序包括:
启用 PL0 和 PL1 第二级内存管理单元;
配置要在管理程序模式下采取中断请求;以及
调用 SMMU 驱动器,以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级

转换内。

13. 根据权利要求 12 所述的方法,还包括与数字信号处理器开始处理器间的通信。

14. 根据权利要求 12 所述的方法,还包括处理 SMMU 故障。

15. 根据权利要求 1 所述的方法,其中,实现访问控制包括实现第二级转换。

16. 根据权利要求 15 所述的方法,其中,实现访问控制还包括下列各项中的至少一项:

重新开始限制所述 HLOS 对硬件中断的访问;

重新开始限制所述 HLOS 对硬件定时器的访问;以及

重新开始限制所述 HLOS 的 I/O 访问。

17. 根据权利要求 15 所述的方法,其中,实现第二级转换包括:

监测由所述 HLOS 分配内存的尝试;以及

在所述 HLOS 尝试分配内存时,向所述 HLOS 提供物理地址空间中对所述 HLOS 可访问的一个或多个物理地址。

18. 根据权利要求 17 所述的方法,还包括:

在所述 HLOS 尝试分配内存时,确定沙盒化的组件是否在尝试分配内存;以及

在确定所述沙盒化的组件在尝试分配内存时,向所述沙盒化的组件提供物理地址,所述物理地址是来自物理地址空间中的物理地址。

19. 根据权利要求 18 所述的方法,其中,向所述沙盒化的组件提供所述物理地址包括:

从所述物理地址空间中对所述 HLOS 可访问的所述物理地址中移除要被提供给所述沙盒化的组件的所述物理地址;以及

向所述沙盒化的组件提供物理地址,所述物理地址是来自所述物理地址空间中的可用的物理地址。

20. 一种计算设备,包括:

内存;以及

处理器,其耦合至所述内存,其中,所述处理器被配置有处理器可执行指令以执行包括下列步骤的操作:

初始化管理程序、安全性监测器以及高级操作系统 (HLOS);

在初始化之后,禁用所述管理程序;

监测来自所述安全性监测器的用于开始沙盒会话的信号;

在所述信号被接收以开始所述沙盒会话时,启用所述管理程序;以及

当所述管理程序被启用时,实现访问控制。

21. 根据权利要求 20 所述的计算设备,其中,所述安全性监测器是 ARM 信任区[®]。

22. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得所述管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或被启用。

23. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得初始化所述管理程序包括配置所述 HLOS 来分配内存空间,使得所述 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。

24. 根据权利要求 23 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得初始化所述管理程序还包括利用所述安全性监测器对所述管理程序的代

码和数据进行认证。

25. 根据权利要求 24 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行还包括下列步骤的操作:当所述管理程序被启用时,配置所述管理程序的代码和数据对数字信号处理器和被包括在所述数字信号处理器中的 CPU 中的至少一个不可访问。

26. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得禁用所述管理程序包括:

配置所有系统内存管理单元(SMMU)的上下文库来绕过第二级转换;以及
针对所述 HLOS 关闭第二级转换。

27. 根据权利要求 26 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得禁用所述管理程序还包括下列各项中的至少一项:

暂停限制所述 HLOS 对硬件中断的访问;
暂停限制所述 HLOS 对硬件定时器的访问;以及
暂停限制所述 HLOS 的 I/O 访问。

28. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行还包括下列步骤的操作:

确定所述沙盒会话是否结束;
在确定所述沙盒会话结束时,执行沙盒会话拆除过程;以及
在执行所述沙盒会话拆除过程之后,禁用所述管理程序。

29. 根据权利要求 28 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得确定所述沙盒会话是否结束包括接收指示所述沙盒会话结束的另一信号。

30. 根据权利要求 28 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得执行所述沙盒会话拆除过程包括:

释用于沙盒化的组件的所有缓冲器;以及
恢复第二级转换页表,以移除所有存储残片。

31. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得启用所述管理程序包括:

启用 PL0 和 PL1 第二级内存管理单元;
配置要在管理程序模式下采取中断请求;以及

调用 SMMU 驱动器,以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级转换内。

32. 根据权利要求 31 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行还包括与数字信号处理器开始处理器间的通信的操作。

33. 根据权利要求 31 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行还包括处理 SMMU 故障的操作。

34. 根据权利要求 20 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得实现访问控制包括实现第二级转换。

35. 根据权利要求 34 所述的计算设备,其中,所述处理器被配置有处理器可执行指令,以执行操作使得实现访问控制还包括下列各项中的至少一项:

重新开始限制所述 HLOS 对硬件中断的访问;

重新开始限制所述 HLOS 对硬件定时器的访问 ; 以及
重新开始限制所述 HLOS 的 I/O 访问。

36. 根据权利要求 34 所述的计算设备, 其中, 所述处理器被配置有处理器可执行指令, 以执行操作使得实现第二级转换包括 :

监测由所述 HLOS 分配内存的尝试 ; 以及

在所述 HLOS 尝试分配内存时, 向所述 HLOS 提供物理地址空间中对所述 HLOS 可访问的一个或多个物理地址。

37. 根据权利要求 36 所述的计算设备, 其中, 所述处理器被配置有处理器可执行指令, 以执行还包括下列步骤的操作 :

在所述 HLOS 尝试分配内存时, 确定沙盒化的组件是否在尝试分配内存 ; 以及

在确定所述沙盒化的组件在尝试分配内存时, 向所述沙盒化的组件提供物理地址, 所述物理地址是来自物理地址空间中的物理地址。

38. 根据权利要求 37 所述的计算设备, 其中, 所述处理器被配置有处理器可执行指令, 以执行操作使得向所述沙盒化的组件提供所述物理地址包括 :

从所述物理地址空间中对所述 HLOS 可访问的所述物理地址中移除要被提供给所述沙盒化的组件的所述物理地址 ; 以及

向所述沙盒化的组件提供物理地址, 所述物理地址是来自所述物理地址空间中的可用的物理地址。

39. 一种计算设备, 包括 :

用于初始化管理程序、安全性监测器以及高级操作系统 (HLOS) 的单元 ;

用于在初始化之后禁用所述管理程序的单元 ;

用于监测来自所述安全性监测器的用于开始沙盒会话的信号的单位 ;

用于在所述信号被接收以开始所述沙盒会话时启用所述管理程序的单元 ; 以及

用于当所述管理程序被启用时实现访问控制的单元。

40. 根据权利要求 39 所述的计算设备, 其中, 所述安全性监测器是 ARM 信任区[®]。

41. 根据权利要求 39 所述的计算设备, 其中, 所述管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或被启用。

42. 根据权利要求 39 所述的计算设备, 其中, 用于初始化所述管理程序的单元包括用于配置所述 HLOS 来分配内存空间, 使得所述 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址的单元。

43. 根据权利要求 42 所述的计算设备, 其中, 用于初始化所述管理程序的单元还包括用于利用所述安全性监测器对所述管理程序的代码和数据认证的单位。

44. 根据权利要求 43 所述的计算设备, 还包括用于当所述管理程序被启用时配置所述管理程序的代码和数据对数字信号处理器和被包括在所述数字信号处理器中的 CPU 中的至少一个不可访问的单元。

45. 根据权利要求 39 所述的计算设备, 其中, 用于禁用所述管理程序的单元包括 :

用于配置所有系统内存管理单元 (SMMU) 的上下文库来绕过第二级转换的单元 ; 以及

用于针对所述 HLOS 关闭第二级转换的单元。

46. 根据权利要求 45 所述的计算设备, 其中, 用于禁用所述管理程序的单元还包括下

列单元中的至少一个单元：

用于暂停限制所述 HLOS 对硬件中断的访问的单元；
用于暂停限制所述 HLOS 对硬件定时器的访问的单元；以及
用于暂停限制所述 HLOS 的 I/O 访问的单元。

47. 根据权利要求 39 所述的计算设备，还包括：

用于确定所述沙盒会话是否结束的单元；
用于在确定所述沙盒会话结束时执行沙盒会话拆除过程的单元；以及
用于在执行所述沙盒会话拆除过程之后禁用所述管理程序的单元。

48. 根据权利要求 47 所述的计算设备，其中，用于确定所述沙盒会话是否结束的单元包括用于接收指示所述沙盒会话结束的另一信号的单元。

49. 根据权利要求 47 所述的计算设备，其中，用于执行所述沙盒会话拆除过程的单元包括：

用于释放用于沙盒化的组件的所有缓冲器的单元；以及
用于恢复第二级转换页表以移除所有存储残片的单元。

50. 根据权利要求 39 所述的计算设备，其中，用于启用所述管理程序的单元包括：

用于启用 PL0 和 PL1 第二级内存管理单元的单元；
用于配置要在管理程序模式下采取中断请求的单元；以及

用于调用 SMMU 驱动器以致使第一级转换中的所有活动的 SMMU 上下文被嵌套在第二级转换内的单元。

51. 根据权利要求 50 所述的计算设备，还包括用于与数字信号处理器开始处理器间的通信的单元。

52. 根据权利要求 50 所述的计算设备，还包括用于处理 SMMU 故障的单元。

53. 根据权利要求 39 所述的计算设备，其中，用于实现访问控制的单元包括用于实现第二级转换的单元。

54. 根据权利要求 53 所述的计算设备，其中，用于实现访问控制的单元还包括下列单元中的至少一个单元：

用于重新开始限制所述 HLOS 对硬件中断的访问的单元；
用于重新开始限制所述 HLOS 对硬件定时器的访问的单元；以及
用于重新开始限制所述 HLOS 的 I/O 访问的单元。

55. 根据权利要求 53 所述的计算设备，其中，用于实现第二级转换的单元包括：

用于监测由所述 HLOS 分配内存的尝试的单元；以及
用于在所述 HLOS 尝试分配内存时，向所述 HLOS 提供物理地址空间中对所述 HLOS 可访问的一个或多个物理地址的单元。

56. 根据权利要求 55 所述的计算设备，还包括：

用于在所述 HLOS 尝试分配内存时，确定沙盒化的组件是否在尝试分配内存的单元；以及

用于在确定所述沙盒化的组件在尝试分配内存时，向所述沙盒化的组件提供物理地址的单元，所述物理地址是来自物理地址空间中的物理地址。

57. 根据权利要求 56 所述的计算设备，其中，用于向所述沙盒化的组件提供所述物理

地址的单元包括：

用于从所述物理地址空间中对所述 HLOS 可访问的所述物理地址中移除要被提供给所述沙盒化的组件的所述物理地址的单元；以及

用于向所述沙盒化的组件提供物理地址的单元，所述物理地址是来自所述物理地址空间中的可用的物理地址。

58. 一种非暂时性处理器可读存储介质，具有被存储在其上的处理器可执行软件指令，所述处理器可执行软件指令被配置为使处理器执行用于管理计算设备上的内存的操作，所述操作包括：

初始化管理程序、安全性监测器以及高级操作系统 (HLOS)；

在初始化之后，禁用所述管理程序；

监测来自所述安全性监测器的用于开始沙盒会话的信号；

在所述信号被接收以开始所述沙盒会话时，启用所述管理程序；以及

当所述管理程序被启用时，实现访问控制。

59. 根据权利要求 58 所述的非暂时性处理器可读存储介质，其中，所述安全性监测器是 ARM 信任区®。

60. 根据权利要求 58 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行操作，使得所述管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或被启用。

61. 根据权利要求 58 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行操作，使得初始化所述管理程序包括配置所述 HLOS 来分配内存空间，使得所述 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。

62. 根据权利要求 61 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行操作，使得初始化所述管理程序还包括利用所述安全性监测器对所述管理程序的代码和数据进行认证。

63. 根据权利要求 62 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行还包括下列步骤的操作：当所述管理程序被启用时，配置所述管理程序的代码和数据对数字信号处理器和被包括在所述数字信号处理器中的 CPU 中的至少一个不可访问。

64. 根据权利要求 58 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行操作，使得禁用所述管理程序包括：

配置所有系统内存管理单元 (SMMU) 的上下文来绕过第二级转换；以及

针对所述 HLOS 关闭第二级转换。

65. 根据权利要求 64 所述的非暂时性处理器可读存储介质，其中，所存储的处理器可执行软件指令被配置为使处理器执行操作，使得禁用所述管理程序还包括下列各项中的至少一项：

暂停限制所述 HLOS 对硬件中断的访问；

暂停限制所述 HLOS 对硬件定时器的访问；以及

暂停限制所述 HLOS 的 I/O 访问。

66. 根据权利要求 58 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行还包括下列步骤的操作:

确定所述沙盒会话是否结束;

在确定所述沙盒会话结束时, 执行沙盒会话拆除过程; 以及

在执行所述沙盒会话拆除过程之后, 禁用所述管理程序。

67. 根据权利要求 66 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得确定所述沙盒会话是否结束包括接收指示所述沙盒会话结束的另一信号。

68. 根据权利要求 66 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得执行所述沙盒会话拆除过程包括:

释用于沙盒化的组件的所有缓冲器; 以及

恢复第二级转换页表, 以移除所有存储残片。

69. 根据权利要求 58 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得启用所述管理程序包括:

启用 PL0 和 PL1 第二级内存管理单元;

配置要在管理程序模式下采取中断请求; 以及

调用 SMMU 驱动器, 以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级转换内。

70. 根据权利要求 69 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行还包括与数字信号处理器开始处理器间的通信的操作。

71. 根据权利要求 69 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行还包括处理 SMMU 故障的操作。

72. 根据权利要求 58 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得实现访问控制包括实现第二级转换。

73. 根据权利要求 72 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得实现访问控制还包括下列各项中的至少一项:

重新开始限制所述 HLOS 对硬件中断的访问;

重新开始限制所述 HLOS 对硬件定时器的访问; 以及

重新开始限制所述 HLOS 的 I/O 访问。

74. 根据权利要求 72 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行操作, 使得实现第二级转换包括:

监测由所述 HLOS 分配内存的尝试; 以及

在所述 HLOS 尝试分配内存时, 向所述 HLOS 提供物理地址空间中对所述 HLOS 可访问的一个或多个物理地址。

75. 根据权利要求 74 所述的非暂时性处理器可读存储介质, 其中, 所存储的处理器可执行软件指令被配置为使处理器执行还包括下列步骤的操作:

在所述 HLOS 尝试分配内存时, 确定沙盒化的组件是否在尝试分配内存; 以及

在确定所述沙盒化的组件在尝试分配内存时,向所述沙盒化的组件提供物理地址,所述物理地址是来自物理地址空间中的物理地址。

76. 根据权利要求 75 所述的非暂时性处理器可读存储介质,其中,所存储的处理器可执行软件指令被配置为使处理器执行操作,使得向所述沙盒化的组件提供所述物理地址包括:

从所述物理地址空间中对所述 HLOS 可访问的所述物理地址中移除要被提供给所述沙盒化的组件的所述物理地址;以及

向所述沙盒化的组件提供物理地址,所述物理地址是来自所述物理地址空间中的可用的物理地址。

按需选择性地启用虚拟机监测器的操作的方法和装置

背景技术

[0001] 通常地,虚拟化技术通过在操作系统与硬件之间放置软件控制程序(例如,虚拟机监测器“VMM”或管理程序(hypervisor))来启用对计算资源的抽象(或虚拟化)。管理程序在特许模式下执行,并且可以存放(host)多个操作系统(被称为客户操作系统)。通过将管理程序和硬件的组合看作单个的虚拟机,每一个客户操作系统以其与物理硬件进行通信的相同的方式与管理程序进行通信。这允许每一个客户操作系统在专有地访问处理器、外围设备、存储器和 I/O 的错觉下进行操作。

[0002] 操作系统负责跨越多个进程划分物理内存。在包括有运行在虚拟机顶部的客户操作系统的系统中,由客户操作系统分配的内存不是真实的物理内存,而是中间物理内存。在这样的系统上,管理程序负责物理内存的实际分配。

[0003] 大部分处理器仅支持一级内存地址空间转换,以及管理程序管理虚拟地址(VA)、中间物理地址(IPA)和物理地址(PA)之间的关系。这通常通过管理程序维护其自己的转换表(被称为影子转换表)来实现,所述转换表是通过解译客户操作系统的转换表中的每一个转换表导出的。具体地,管理程序确保对客户操作系统的转换表的所有改变被反映在影子结构中,以及实施保护并将访问故障重定向到恰当的级。

[0004] 不像上文讨论的单级处理器,ARM 处理器系统(例如,经由诸如系统内存管理单元“SMMU”之类的 ARM 虚拟化扩展)为两级内存转换提供硬件协助。例如,ARM 处理器包括启用两级转换的虚拟化扩展,在所述两级转换中,在第一级(即,第一级转换)中,将虚拟地址(VA)转换至中间物理地址(IPA),以及在第二级(即,第二级转换)中,将中间物理地址(IPA)转换至物理地址。这降低了与管理程序相关联的开销。

发明内容

[0005] 各个方面包括用于选择性地实现管理程序来在需要的时候有效地实施访问控制,以保护数据和/或软件的计算设备和方法。在各个方面,管理程序通常被禁用,并且在检测到需要管理程序来实现访问控制(即,“沙盒会话”)的状况时被启用。高级操作系统(HLOS)可以在由管理程序管理的虚拟机中操作。HLOS 可以维护中间物理地址页表,以用于将虚拟地址分配给运行在 HLOS 上的各个进程或应用。HLOS 可以直接从物理内存地址空间分配内存,由此确保中间物理内存地址总是与物理内存地址相同。通过确保 HLOS 能够分配内存,以使中间物理地址一直等于物理地址,管理程序可以在存在针对沙盒化的会话的需求时被选择性地启用,以及在当前不存在沙盒化的会话时被禁用。当被禁用时,管理程序不可以执行从中间物理地址到物理地址的级 2 转换。同样地,当管理程序被禁用时,HLOS 可以从整个物理内存地址空间来分配内存。

[0006] 在各个方面,管理程序可以在沙盒会话期间被启用。当被启用时,管理程序可以重新开始执行从中间物理地址到物理地址的级 2 转换。同样地,当被启用并且管理程序可以限制 HLOS 对物理内存地址空间的访问时,由此允许 HLOS 仅从物理内存地址空间的一部分分配内存,以及在一些方面,限制 HLOS 对硬件中断和/或硬件定时器的访问。当不需要沙

盒时（即，当管理程序被禁用时），除了别的之外，通过配置管理程序不执行级 2 转换，各个方面可以在按照需要提供必要的安全性的同时改善计算设备的整体性能。

[0007] 各个方面包括一种通过下述步骤来管理计算设备上的内存的方法，包括：初始化管理程序、安全性监测器以及高级操作系统（HLOS）；在初始化之后，禁用管理程序；监测来自安全性监测器的用于开始沙盒会话的信号；在信号被接收以开始沙盒会话时启用管理程序；以及当管理程序被启用时实现访问控制。在一个方面，安全性监测器可以是 ARM 信任区[®]。在另一方面，管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或启用。在另一方面，初始化管理程序可以包括配置 HLOS 来分配内存空间，以使 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。在又一方面，初始化管理程序还可以包括利用安全性监测器对管理程序的代码和数据进行认证。在另一方面，该方法可以包括当管理程序被启用时，配置管理程序的代码和数据对数字信号处理器和被包括在数字信号处理器中的 CPU 中的至少一个不可访问。

[0008] 在另一方面，禁用管理程序可以包括配置所有系统内存管理单元（SMMU）的上下文库来绕过第二级转换，并针对 HLOS 关闭第二级转换。在一个方面，禁用管理程序可以包括暂停限制 HLOS 对硬件中断的访问、暂停限制 HLOS 对硬件定时器的访问以及暂停限制 HLOS 的 I/O 访问中的至少一个。

[0009] 在另一方面，该方法可以包括确定沙盒会话是否结束，在确定沙盒会话结束时执行沙盒会话拆除过程，以及在执行沙盒会话拆除过程之后禁用管理程序。在另一方面，确定沙盒会话是否结束可以包括接收指示沙盒会话结束的另一信号。在又一方面，执行沙盒会话拆除过程可以包括释放用于沙盒化的组件的所有缓冲区以及恢复第二级转换页表，以移除所有的存储残片。

[0010] 在一个方面，启用管理程序可以包括启用 PL0 和 PL1 第二级内存管理单元，配置要在管理程序模式下采取中断请求，以及调用 SMMU 驱动器以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级转换内。在另一方面，该方法还可以包括与数字信号处理器开始处理器间的通信。在另一方面，该方法还可以包括处理 SMMU 故障。

[0011] 在另一方面，实现访问控制可以包括实现第二级转换。在又一方面，实现访问控制可以包括重新开始限制 HLOS 对硬件中断的访问，重新开始限制 HLOS 对硬件定时器的访问以及重新开始限制 HLOS 的 I/O 访问中的至少一个。在一个方面，实现第二级转换可以包括监测由 HLOS 分配内存的尝试，以及在 HLOS 尝试分配内存时，向 HLOS 提供物理地址空间中对 HLOS 可访问的一个或多个物理地址。

[0012] 在另一方面，该方法可以包括在 HLOS 尝试分配内存时确定沙盒化的组件是否在尝试分配内存，以及在确定沙盒化的组件在尝试分配内存时向沙盒化的组件提供物理地址，所述物理地址是来自物理地址空间中的物理地址。在一个方面，向沙盒化的组件提供物理地址可以包括从物理地址空间中对 HLOS 可访问的物理地址中移除要被提供给沙盒化的组件的物理地址，以及向沙盒化的组件提供物理地址，所述物理地址是来自物理地址空间中的可用的物理地址。

[0013] 另外的方面包括一种计算设备，其可以包括内存和耦合至内存的处理器，其中，该处理器被配置有处理器可执行指令，以执行包括下述的操作：初始化管理程序、安全性监测器以及高级操作系统（HLOS），在初始化之后禁用管理程序，监测来自安全性监测器的用于

开始沙盒会话的信号,在信号被接收以开始沙盒会话时启用管理程序,以及当管理程序被启用时实现访问控制。在另一方面,安全性监测器可以是 ARM 信任区[®]。在另一方面,处理器可以被配置有处理器可执行指令,以执行操作使得管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或启用。

[0014] 在又一方面,处理器可以被配置有处理器可执行指令,以执行操作使得初始化管理程序包括配置 HLOS 来分配内存空间,使得 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。在一个方面,处理器可以被配置有处理器可执行指令,以执行操作使得初始化管理程序包括利用安全性监测器对管理程序的代码和数据进行认证。在另一方面,处理器可以被配置有处理器可执行指令,以执行包括下述的操作:当管理程序被启用时,配置管理程序的代码和数据对数字信号处理器和被包括在数字信号处理器中的 CPU 中的至少一个不可访问。

[0015] 在一个方面,处理器可以被配置有处理器可执行指令,以执行操作使得禁用管理程序包括配置所有系统内存管理单元(SMMU)的上下文库来绕过第二级转换,以及针对 HLOS 关闭第二级转换。在另一方面,处理器可以被配置有处理器可执行指令,以执行操作使得禁用管理程序还包括暂停限制 HLOS 对硬件中断的访问,暂停限制 HLOS 对硬件定时器的访问以及暂停限制 HLOS 的 I/O 访问中的至少一个。

[0016] 在另一方面,处理器可以被配置有处理器可执行指令,以执行包括下述的操作:确定沙盒会话是否结束,在确定沙盒会话结束时执行沙盒会话拆除过程,以及在执行沙盒会话拆除过程之后禁用管理程序。在另一方面,处理器可以被配置有处理器可执行指令,以执行操作使得确定沙盒会话是否结束包括接收指示沙盒会话结束的另一信号。在又一方面,处理器可以被配置有处理器可执行指令,以执行操作使得执行沙盒会话拆除过程包括释用于沙盒化的组件的所有缓冲器以及恢复第二级转换页表,以移除所有存储残片。

[0017] 在一个方面,处理器可以被配置有处理器可执行指令,以执行操作使得启用管理程序包括:启用 PL0 和 PL1 第二级内存管理单元,配置要在管理程序模式下采取中断请求,以及调用 SMMU 驱动器以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级转换内。在另一方面,处理器可以被配置有处理器可执行指令,以执行包括与数字信号处理器开始处理器间的通信的操作。在又一方面,处理器可以被配置有处理器可执行指令,以执行包括处理 SMMU 故障的操作。

[0018] 在一个方面,处理器可以被配置有处理器可执行指令,以执行操作使得实现访问控制包括实现第二级转换。在另一方面,处理器可以被配置有处理器可执行指令,以执行操作使得执行访问控制还包括重新开始限制 HLOS 对硬件中断的访问,重新开始限制 HLOS 对硬件定时器的访问以及重新开始限制 HLOS 的 I/O 访问中的至少一个。

[0019] 在一个方面,处理器可以被配置有处理器可执行指令,以执行操作使得执行第二级转换包括监测由 HLOS 分配内存的尝试,以及在 HLOS 尝试分配内存时向 HLOS 提供物理地址空间中对 HLOS 可访问的一个或多个物理地址。在另一方面,处理器可以被配置有处理器可执行指令,以执行还包括下述的操作:在 HLOS 尝试分配内存时,确定沙盒化的组件是否在尝试分配内存,以及在确定沙盒化的组件在尝试分配内存时,向沙盒化的组件提供物理地址,所述物理地址是来自物理地址空间中的物理地址。在又一方面,处理器可以被配置有处理器可执行指令,以执行操作使得向沙盒化的组件提供物理地址包括从物理地址空间中

对 HLOS 可访问的物理地址中移除要被提供给沙盒化的组件的物理地址,以及向沙盒化的组件提供物理地址,所述物理地址是来自物理地址空间中的可用的物理地址。

[0020] 另外的方面包括一种计算设备,包括用于初始化管理程序、安全性监测器和高级操作系统 (HLOS) 的单元;用于在初始化之后禁用管理程序的单元;用于监测来自安全性监测器的用于开始沙盒会话的信号;用于在信号被接收以开始沙盒会话时启用管理程序的单元;以及用于当管理程序被启用时实现访问控制的单元。在另一方面,安全性监测器可以是 ARM 信任区[®]。在另一方面,管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或启用。在另一方面,用于初始化管理程序的单元可以包括用于配置 HLOS 来分配内存空间,使得 HLOS 的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址的单元。在另一方面,用于初始化管理程序的单元还可以包括用于利用安全性监测器对管理程序的代码和数据认证认证的单元。在又一方面,计算设备可以包括用于当管理程序被启用时,配置管理程序的代码和数据对数字信号处理器和被包括在数字信号处理器中的 CPU 中的至少一个不可访问的单元。

[0021] 在一个方面,用于禁用管理程序的单元可以包括用于配置所有系统内存管理单元 (SMMU) 的上下文库来绕过第二级转换的单元,以及用于针对 HLOS 关闭第二级转换的单元。在另一方面,用于禁用管理程序的单元还可以包括下述单元中的至少一个单元:用于暂停限制 HLOS 对硬件中断的访问的单元,用于暂停限制 HLOS 对硬件定时器的访问的单元以及用于暂停限制 HLOS 的 I/O 访问的单元。

[0022] 在一个方面,计算设备还可以包括用于确定沙盒会话是否结束的单元,用于在确定沙盒会话结束时执行沙盒会话拆除过程的单元,以及用于在执行沙盒会话拆除过程之后禁用管理程序的单元。在另一方面,用于确定沙盒会话是否结束的单元可以包括用于接收指示沙盒会话结束的另一信号的单元。在另一方面,用于执行沙盒会话拆除过程的单元可以包括用于释放用于沙盒化的组件的所有缓冲器的单元,以及用于恢复第二级转换页表以移除所有存储残片的单元。

[0023] 在一个方面,用于启用管理程序的单元可以包括用于启用 PL0 和 PL1 第二级内存管理单元的单元,用于配置要在管理程序模式下采取中断请求的单元,以及用于调用 SMMU 驱动器以致使第一级转换中的所有活动的 SMMU 上下文库被嵌套在第二级转换内。在另一方面,计算设备可以包括用于与数字信号处理器开始处理器间通信的单元。在另一方面,计算设备可以包括用于处理 SMMU 故障的单元。

[0024] 在一个方面,用于实现访问控制的单元可以包括用于实现第二级转换的单元。在另一方面,用于实现访问控制的单元可以包括下述单元中的至少一个单元:用于重新开始限制 HLOS 对硬件中断的访问的单元,用于重新开始限制 HLOS 对硬件定时器的访问的单元以及用于重新开始限制 HLOS 的 I/O 访问的单元。在另一方面,用于实现第二级转换的单元可以包括用于监测由 HLOS 分配内存的尝试的单元,以及用于在 HLOS 尝试分配内存时向 HLOS 提供物理地址空间中对 HLOS 可访问的一个或多个物理地址的单元。在另一方面,计算设备还可以包括用于在 HLOS 尝试分配内存时确定沙盒化的组件是否在尝试分配内存的单元,以及用于在确定沙盒化的组件在尝试分配内存时向沙盒化的组件提供物理地址的单元,所述物理地址是来自物理地址空间中的物理地址。在另一方面,用于向沙盒化的组件提供物理地址的单元可以包括用于从物理地址空间中对 HLOS 可访问的物理地址中移除要被

提供给沙盒化的组件的物理地址的单元,以及用于向沙盒化的组件提供物理地址的单元,所述物理地址是来自物理地址空间中的可用的物理地址。

[0025] 在又一方面,非暂时性处理器可读存储介质可以具有被存储在其上的处理器可执行软件指令,所述处理器可执行软件指令被配置为使处理器执行用于管理计算设备上的内存的操作,所述操作包括初始化管理程序、安全性监测器和高级操作系统(HLOS);在初始化之后禁用管理程序;监测来自安全性监测器的用于开始沙盒会话的信号;在信号被接收以开始沙盒会话时启用管理程序;以及当管理程序被启用时实现访问控制。在另一方面,安全性监测器可以是ARM信任区[®]。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得管理程序可以跨越集成电路边界和芯片边界中的至少一个被禁用或启用。

[0026] 在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得初始化管理程序包括配置HLOS来分配内存空间,使得HLOS的中间物理地址空间中的每一个中间物理地址等于物理地址空间中相应的物理地址。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得初始化管理程序包括利用安全性监测器对管理程序的代码和数据进行认证。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行包括下述的操作:当管理程序被启用时,配置管理程序的代码和数据对数字信号处理器和被包括在数字信号处理器中的CPU中的至少一个不可访问。

[0027] 在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得禁用管理程序包括配置所有系统内存管理单元(SMMU)的上下文库绕过第二级转换,以及针对HLOS关闭第二级转换。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得禁用管理程序还包括暂停限制HLOS对硬件中断的访问,暂停限制HLOS对硬件定时器的访问,以及暂停限制HLOS的I/O访问中的至少一个。

[0028] 在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行包括下述的操作:确定沙盒会话是否结束,在确定沙盒会话结束时执行沙盒会话拆除过程,以及在执行沙盒会话拆除过程之后禁用管理程序。在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得确定沙盒是否结束包括接收指示沙盒结束的另一信号。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得执行沙盒会话拆除过程包括释放用于沙盒化的组件的所有缓冲器,以及恢复第二级转换页表以移除所有存储残片。

[0029] 在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得启用处理器包括启用PL0和PL1第二级内存管理单元,配置要在管理程序模式下采取中断请求,以及调用SMMU驱动器以致使第一级转换中的所有活动的SMMU上下文库被嵌套在第二级转换内。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行包括与数字信号处理器开始处理器间通信的操作。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行包括处理SMMU故障的操作。

[0030] 在一个方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得实现访问控制包括实现第二级转换。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得实现访问控制还包括重新开始限制HLOS对硬件中断的访问,重新开始限制HLOS对硬件定时器的访问,以及重新开始限制HLOS的I/O访问中的至少

一个。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得实现第二级转换包括监测由 HLOS 分配内存的尝试,以及在 HLOS 尝试分配内存时向 HLOS 提供物理地址空间中对 HLOS 可访问的一个或多个物理地址。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行包括下述的操作:在 HLOS 尝试分配内存时,确定沙盒化的组件是否在尝试分配内存,以及在确定沙盒化的组件在尝试分配内存时,向沙盒化的组件提供物理地址,所述物理地址是来自物理地址空间中的物理地址。在另一方面,存储的处理器可执行软件指令可以被配置为使处理器执行操作,使得向沙盒化的组件提供物理地址包括从物理地址空间中对 HLOS 可访问的物理地址中移除要被提供给沙盒化的组件的物理地址,以及向沙盒化的组件提供物理地址,所述物理地址是物理地址空间中的可用的物理地址。

附图说明

[0031] 被并入本文并构成本说明书的一部分的附图示出本发明的示例性方面,并且与上文给出的一般描述和下文给出的具体实施方式一起用来解释本发明的特征。

[0032] 图 1 是一个方面的计算设备的组件框图。

[0033] 图 2 是计算设备的模块的功能框图。

[0034] 图 3 是一个方面的计算系统的分层的计算机架构图。

[0035] 图 4 和 5 是虚拟机中一个方面的逻辑组件的分层的计算机架构图。

[0036] 图 6 是示出实现系统虚拟机的计算设备中的两级内存地址映射的功能框和内存映射图。

[0037] 图 7 是示出当管理程序被禁用时实现系统虚拟机的计算设备中的两级内存地址映射的内存映射图。

[0038] 图 8 是示出在沙盒会话期间当管理程序被启用时实现系统虚拟机的计算设备中的两级内存地址映射的内存映射图。

[0039] 图 9 是示出在沙盒会话期间当管理程序被启用时实现具有共享虚拟内存的系统虚拟机的计算设备中的两级内存地址映射的内存映射图。

[0040] 图 10 是示出用于选择性地启用和禁用管理程序的一个方面的方法的过程流程图。

[0041] 图 11 是示出涉及在计算设备上发起共享虚拟内存环境的信令的调用流程图。

[0042] 图 12 是示出配置共享虚拟内存会话的一个方面的方法的过程流程图。

[0043] 图 13 是示出禁用管理程序的一个方面的方法的过程流程图。

[0044] 图 14 是示出涉及在计算设备上发起相应的沙盒会话的信令的调用流程图。

[0045] 图 15 是示出启用管理程序的一个方面的方法的过程流程图。

[0046] 图 16A 和图 16B 是示出实现第二级转换的一个方面的方法的过程流程图。

[0047] 图 17 是示出执行沙盒会话拆除的一个方面的方法的过程流程图。

[0048] 图 18 是适合于实现各个方面的计算设备的组件框图。

[0049] 图 19 是适合于实现各个方面的另一计算设备的组件框图。

具体实施方式

[0050] 将关于附图详细地描述各个方面。只要有可能,遍及附图将使用相同的参考号来指代相同或相似的元器件。对特定示例和实现方式进行的参考是出于说明性的目的的,并不旨在限定本发明或权利要求的范围。

[0051] 在综述中,各个方面包括一种计算设备和用于在计算设备上选择性地实现管理程序以有效地实施关于管理程序的访问控制的方法。在各个方面,管理程序通常被禁用,以及在检测到将需要管理程序来实现访问控制(即,“沙盒会话”)的情况时被启用。当被禁用时,管理程序可能不会执行第二级转换,但是,当被启用时,管理程序可以重新开始第二级转换。在一些方面,当被启用时,管理程序还可以重新开始当管理程序被禁用时暂停的其它活动,包括限制输入/输出(I/O)访问、硬件中断访问和/或硬件定时器访问。因此,仅当需要沙盒时通过选择性地启用管理程序,计算设备可以改善整个性能和用户体验,同时维护安全的操作环境。

[0052] 在各个方面,计算设备可以包括内存和被耦合至内存的处理器,所述处理器被配置有管理程序,被实现在硬件(即,裸金属的管理程序)中、软件(即,在传统的操作系统环境内操作的群集的管理程序)中或硬件与软件的组合中。管理程序另外可以为各个应用、操作系统、进程、信号等生成并管理一个或多个虚拟机。

[0053] 本文使用术语“计算设备”来指代蜂窝电话、智能手机、个人的或移动的多媒体播放器、个人数字助理(PDA的)、膝上型电脑、平板电脑、智能本、掌上型电脑、无线电子邮件接收器、启用多媒体互联网的蜂窝电话、无线游戏控制器以及包括可编程处理器和内存的类似的个人电子设备,以及在电池电源下操作,以便受益于省电的方法。虽然各个方面在具有有限的处理能力和电池容量的移动设备(诸如蜂窝电话)中尤其有用,但是,概括地说,这些方面在可以受益于改善的处理器性能和降低的能耗的任意计算设备中有用。

[0054] 本文可互换地使用术语“虚拟机监测器”、“VMM”和“管理程序”来指代虚拟机管理器。本文使用术语“高级操作系统”(HLOS)来指代操作在管理程序管理的虚拟机中的客户操作系统。在一个方面,管理程序可以在沙盒会话期间隔离HLOS。

[0055] 本文可互换地使用术语“级1转换”和“第一级转换”来指代从虚拟内存地址(“VA”)到中间内存地址(“IPA”)的转换或映射。在示例中,HLOS可以执行从分配给运行在HLOS上的进程的虚拟地址到HLOS的中间物理地址空间中维护的中间物理地址的第一级转换。

[0056] 本文可互换地使用术语“级2转换”和“第二级转换”来指代从中间物理地址到物理内存地址(“PA”)的转换或映射。在示例中,管理程序或系统内存管理单元(“SMMU”)可以执行从分配给HLOS的中间物理地址到由物理地址空间中的管理程序维护的物理地址的第二级转换。

[0057] 本文使用术语“沙盒会话”来指代在其中管理程序在执行两个或更多个实体之间的访问控制的一段时间。在一个方面,沙盒会话可以在管理程序被警告受保护的内容(例如,由数字版权管理(DRM)技术保护的内容)必须在计算设备(例如,用于播放DRM媒体的安全视频设备)上被单独地处理时开始,以及可以在那个分离不再是必须的时(诸如,当已经完成对受保护的内容的处理或播放,以及服务或应用已经被释放时)结束。

[0058] 本文使用术语“沙盒化的组件”来指代管理程序从HLOS分离(即,沙盒)的组件、应用、进程等。在一个方面,管理程序可以为沙盒化的组件分配物理内存地址空间中不与被

分配给 HLOS 的物理地址重叠的物理地址。

[0059] 本文使用术语“共享实体”来指代与 HLOS 共享虚拟内存的组件、应用、进程等。在一个方面，共享实体和 HLOS 可以共享对物理内存地址空间中的一个或多个物理地址的访问。

[0060] 通过生成和管理虚拟机，管理程序可以在各个操作或数据（包括操作系统、应用、进程等）周围生成“沙盒”或安全的隔离。管理程序可以使用沙盒来限制对各个特征的访问，由此为操作或数据提供安全性。例如，HLOS 可以操作为管理程序管理的虚拟机内的客户操作系统，以及管理程序可以管理在 HLOS 的虚拟机外部处理的视频信号，以使 HLOS 可能不知道（即，不能检测或访问）该视频信号。

[0061] 然而，存在与使用管理程序来实施访问控制相关联的性能开销。基准测试已经表明，取决于基准，使用管理程序来实施访问控制可以使性能下降大约 5% 至 30%。各个方面通过仅在针对由沙盒提供的数据和 / 或软件安全存在需求时实现管理程序来克服这个性能代偿。

[0062] 在各个方面，HLOS 可以在由管理程序管理的虚拟机中操作。HLOS 可以维护中间物理地址页表，供在向运行在 HLOS 上的各个进程或应用分配虚拟地址时使用。HLOS 可以直接从物理内存地址空间分配内存，由此确保中间物理地址总是与物理地址相同。换句话说，HLOS 的中间物理地址空间中的中间物理地址总是等于物理地址空间中的物理地址。通过确保 HLOS 能够分配内存，使得中间物理地址总是等于物理地址，各个方面使得管理程序能够被选择性地启用和禁用，由此，因为仅在需要沙盒会话时引发管理程序的性能影响而改善整个性能。

[0063] 在各个方面，HLOS 可以直接从物理内存地址空间分配内存。然而，当管理程序被启用时，管理程序可以限制 HLOS 对物理内存地址空间的访问，由此允许 HLOS 仅从物理内存地址空间的一部分分配内存。

[0064] 在一个方面，计算设备处理器可以监测需要沙盒化的情况（即，数据和 / 或软件应当在沙盒会话中被处理）。在一个方面，沙盒会话可以是在其中需要进行沙盒来实施单独的进程、应用等之间的访问控制的情况或一段时间。例如，计算设备可以检测已经接收到安全信号（例如，受到数字版权管理的视频信号），或者在发起必须与 HLOS 保持隔离的第二操作系统。在检测到针对沙盒会话的需求时，计算设备可以启用管理程序。在被启用之后，管理程序可以实现第二级转换，并限制 HLOS 仅从物理内存地址空间的一部分分配内存，由此建立沙盒会话。管理程序当被启用时，还可以重新开始其它访问控制操作，诸如，限制 I/O 访问、硬件中断访问和硬件定时器访问中的一个或多个。

[0065] 在另一方面，管理程序可以监测沙盒会话的结束，例如，管理程序可以确定安全的视频信号是否不再被接收。当沙盒会话已经结束时，管理程序可以执行会话拆除。在会话拆除进程中，管理程序可以释放被分配给沙盒化的组件的资源。例如，管理程序可以释放被分配给数字信号处理器来处理安全的视频信号的资源，由此允许 HLOS 从整个物理内存地址空间分配内存。在又一方面，管理程序可以被禁用，直到下一个沙盒情况开始为止。

[0066] 在另一方面，管理程序可以针对 HLOS 和共享实体启用第二级转换。在这个方面，共享实体和 HLOS 可以共享虚拟内存，并共享对物理内存地址空间中的物理地址的访问。在又一方面，管理程序可以针对不被共享的内存地址执行 HLOS 与共享实体之间的访问控制。

[0067] 可以在各种各样的单处理器或多处理器的计算机架构上实现各个方面,图 1 中示出了所述计算机架构的示例。计算设备 100 可以包括多个异构处理器,诸如,示出的数字信号处理器 (DSP) 102、调制解调器处理器 104、图形处理器 106 和应用处理器 108。计算设备 100 还可以包括被连接至处理器 102-108 中的一个或多个处理器的一个或多个向量协处理器 110。每一个处理器 102-110 可以包括一个或多个内核,以及每一个处理器 / 内核可以独立于其它处理器 / 内核执行操作。每一个处理器 102-110 还可以包括内存 (未示出) 和 / 或内存管理系统控制器。在一个方面,计算设备 100 的组件可以位于单个衬底上和 / 或作为片上系统 (SOC) 125 紧密地耦合在一起。

[0068] 计算设备 100 可以包括用于管理传感器数据、模数转换、无线数据传输以及执行其它专用操作 (诸如,针对游戏和电影处理经编码的音频信号) 的模拟电路和定制电路 114。计算设备 100 还可以包括系统组件和资源 116,诸如,电压调节器、振荡器、锁相环、外围的桥接器、数据控制器、内存控制器、系统控制器、访问端口、定时器以及被用来支持处理器、内存和运行在计算设备上的客户端的其它类似的组件。系统组件 / 资源 116 中的每一个还可以包括内存 (未示出) 和 / 或内存管理系统控制器。

[0069] 在各个方面,应用处理器 108 可以是中央处理单元 (CPU)、CPU 的组件或被耦合至 CPU 的处理单元。在一个方面,CPU 可以被配置为从处理器 102-110、系统组件 / 资源 116 和 / 或外围设备的各个内存读取信息,以及向处理器 102-110、系统组件 / 资源 116 和 / 或外围设备的各个内存写入信息,这可以经由相应的处理器 102-110、资源 116 和 / 或外围设备的内存管理系统控制器来实现。

[0070] 计算系统 100 还可以包括用于组件与资源 (诸如,时钟 118 和电压调节器 120) 之间的通信的输入 / 输出模块 (未示出)。可以经由互连 / 总线模块 122 将处理器 102-108 互连至一个或多个存储元件 112、资源 116、定制电路 114 和各个其它系统组件。

[0071] 如上文提到的,计算设备 100 可以包括被连接至处理器 102-108 中的一个或多个处理器的一个或多个向量协处理器 110。这样的向量协处理器 110 对于处理需要快速且并行执行的应用 (诸如,多媒体应用和视频流应用) 尤其有用。在一个方面,向量协处理器 110 可以实现包括独立的硬件寄存器、内存和 / 或执行硬件的单指令多数据 (SIMD) 指令集架构 (ISA)。SIMD 向量协处理器可以是计算设备 100 的主处理器 (例如,应用处理器 108、CPU 等) 的一部分,或者被紧密地耦合至计算设备 100 的主处理器。

[0072] 图 2 示出能够维护安全的虚拟环境 (即,沙盒) 的一个方面的计算设备 206 的组件图。不安全的操作系统 208 (即,HLOS) 可以与管理程序 212 相通信。管理程序可以与物理内存 216 相通信。在一个方面,管理程序可以充当不安全的操作系统 208 与物理内存 216 或其它硬件 (未示出) 之间的媒介物。在另一方面,管理程序 212 可以有助于由不安全的操作系统 208 维护的中间物理地址 (IPA) 到物理内存 216 中的物理地址 (PA) 的映射。

[0073] 在一个方面,管理程序 212 还可以与安全性监测器 (例如,ARM 信任区®) 相通信。安全性监测器 214 可以充当网守,确保仅安全的数据进入和离开安全的虚拟环境 210。安全的虚拟环境 210 继而可以与安全的网络 204 相通信。安全的虚拟环境 210 可以向安全的网络 204 发送数据,或者从安全的网络 204 接收数据。在示例中,安全的虚拟环境 210 可以包括可以从安全的网络 204 接收敏感数据的数字信号处理器 (即,DSP)。在本示例中,敏感数据可以是包含由数字版权管理限制调控的视频数据的信号。在一个方面,安全性监测器 214

可以与管理程序 212 相通信,以确保敏感数据被存储在不可访问不安全的操作系统(或其它系统或进程)的物理内存 216 的一部分中。在又一方面,该敏感数据可以被存储在物理内存 216 内被加密的内存(未示出)中。

[0074] 图 3 示出了示出典型的计算机系统中的逻辑组件和接口的处理器的分层架构。示出的计算机系统架构 300 既包括硬件组件 322 又包括软件组件 320。软件组件 320 可以包括操作系统 302、库模块 304 和一个或多个应用程序 (A_0 至 A_n) 306。硬件组件 322 可以包括外围设备 308(例如,硬件加速器、输入/输出设备等)、中央处理单元 (CPU) 310、中央处理单元内存管理单元 (CPU MMU) 316、一个或多个系统内存管理单元(本文的“系统 MMU”或“SMMU”) 312 以及一个或多个内存 314。

[0075] 为移动计算设备编写的应用软件可以被编译成可执行代码,其是通常被称为“应用(application)”、“应用软件(app)”或应用程序 306 的代码。每一个应用程序 306 可以是单个进程或线程,或者可以包括多个进程或线程。

[0076] 应用程序 306 可以经由应用程序接口 (API) 向库模块 304 发出高级语言 (HLL) 库调用。库模块 304 可以经由应用二进制接口 (ABI) 调用操作系统 302 上的服务(例如,经由操作系统调用)。操作系统 302 可以使用专用指令集架构 (ISA) 来与硬件组件相通信,所述专用指令集架构是由硬件 322 实现的专用操作码 (opcode) 和本机命令的列表。以这种方式,指令集架构可以定义如被操作系统 302 看见的硬件 322。

[0077] 操作系统 302 可以负责协调和控制对应用程序 306 之间的各个内存 314 的分配和使用,其可以包括跨越多个应用程序 (A_0 - A_n) 306 划分物理内存。在一个方面,操作系统 302 可以包括用于通过各个应用程序 (A_0 至 A_n) 306 来管理对系统内存的分配和使用的一个或多个内存管理系统(例如,虚拟内存管理等)。内存管理系统可以起作用以确保被一个进程使用的内存不与已经被另一进程在用的内存冲突。

[0078] 在一个方面,操作系统 302 可以包括虚拟内存管理器 (OS VMM),其被配置为执行使得操作系统 302 能够使特定物理地址似乎是另一地址(即,虚拟地址)的“虚拟寻址”操作。虚拟寻址操作可以包括向应用程序 (A_0 - A_n) 306 分配虚拟内存地址。

[0079] 除上文讨论的基于软件的内存管理系统(例如,OS VMM 等)之外,该系统可以包括一个或多个基于硬件的内存管理系统,诸如,图 3 中示出的中央处理单元 (CPU) 内存管理单元 (MMU) 316 和系统 MMU 312。CPU MMU 316 和系统 MMU 312 均可以包括负责执行与各个内存有关的操作(例如,虚拟地址至物理地址的转换、超高速缓冲器控制、总线仲裁以及内存保护)的一个或多个硬件组件。在一个方面,CPU MMU 316 可以负责向主 CPU 310 提供地址转换服务和保护功能,以及系统 MMU 312 可以负责向其它硬件组件(例如,数字信号处理器、调制解调器处理器、图形处理器等)提供地址转换服务和保护功能。

[0080] 在各个方面,内存管理系统中的一个或多个内存管理系统(例如,系统 MMU 312、CPU MMU 316 等)可以包括传输后备缓冲器 (TLB),其是可以被用于内存地址转换(例如,将虚拟地址转换至物理地址等)的高速缓冲存储器。在一个方面,传输后备缓冲器 (TLB) 可以是内容可寻址存储器 (CAM),其可以是在其中存储的信息被组织成键 (key) 值格式(例如,哈希表)的硬件相联阵列存储器。该键可能是虚拟地址,而该值可能是物理地址。在各个方面,传输后备缓冲器可以是被硬件管理的、被软件管理的或被硬件与软件的组合管理的。关于被硬件管理的传输后备缓冲器,传输后备缓冲器的条目的格式对软件可以是不可

见的,并且因此针对不同类型的中央处理单元可以是不同的。

[0081] 通常地,作为内存地址转换过程的一部分,内存管理系统(例如,OS VMM、系统 MMU 312、CPU MMU 316 等)可以执行内容可寻址内存搜索,以通过向传输后备缓冲器发送作为键的虚拟地址来从传输后备缓冲器请求物理地址。如果在传输后备缓冲器中虚拟地址键具有相应的物理地址(即,发生“TLB 命中”),则内容可寻址内存搜索可以检索并返回相应的物理地址。如果请求的地址不在传输后备缓冲器中(即,发生“TLB 错过”),则内存地址转换过程可以通过读取多个内存位置的内容并计算物理地址来执行页移动(例如,软件页移动、硬件页移动等)。在通过页移动确定物理地址之后,可以将虚拟地址至物理地址的映射存储在传输后备缓冲器中。

[0082] 在包括被软件管理的传输后备缓冲器的方面,TLB 错过可以使操作系统在软件中移动页表并执行转换。在包括被硬件管理的传输后备缓冲器的方面,内存管理系统可以执行硬件表移动,以确定针对指定的虚拟地址键是否存在有效的页表条目。

[0083] 各个方面提供利用虚拟化技术的内存管理系统。虚拟化技术使能计算资源的抽象(或虚拟化),这可以通过在操作系统与硬件之间放置控制程序(例如,虚拟机监测器“VMM”或管理程序)来实现。虚拟化技术通常被实现在虚拟机(VM)中,所述虚拟机可以是像物理硬件机一样执行应用程序的软件应用。虚拟机在应用程序与执行硬件之间提供接口,允许捆绑到专用指令集架构的应用程序在实现不同指令集架构的硬件上执行。

[0084] 图 4 和 5 示出实现虚拟机的典型的计算机系统逻辑组件。可以将虚拟机分成两个一般的类别:系统虚拟机;以及进程虚拟机。系统虚拟机允许不同进程或应用之间的底层物理硬件的共享。另一方面,进程虚拟机支持单个进程或应用。

[0085] 图 4 是示出实现进程虚拟机 410 的计算设备 400 的逻辑层的分层架构图。计算机系统 400 可以包括硬件 408 和包括应用进程模块 402、虚拟化模块 404 和操作系统 406 的软件组件。

[0086] 如上文关于图 3 讨论的,硬件组件通过操作系统 302 仅对应用程序 306 可见,以及 ABI 和 API 有效地定义可用于应用程序 306 的硬件特征。虚拟化软件模块 404 可以在 ABI/API 层面处执行逻辑操作,和/或模仿操作系统调用或库调用,以便应用进程 402 以其以别的方式(即,经由系统调用/库调用)与硬件组件相通信的相同的方式与虚拟化软件模块 404 相通信。以这种方式,应用进程 402 将虚拟化模块 404、操作系统 406 和硬件的组合看作单机,诸如,图 4 中示出的进程虚拟机 410。这简化了应用开发者的工作,因为应用软件不需要关心应用将最终在其上执行的计算设备的真实架构。

[0087] 进程虚拟机 410 单独地存在,以支持单个应用进程 402,并且因此关于进程 402 被生成,并且当进程 402 完成执行时被终止。运行在虚拟机 410 上的进程 402 被称为“客”,以及底层平台被称为“主”。实现进程虚拟机的虚拟化软件 404 典型地被称为运行时间软件(或简单地“运行时间”)。

[0088] 图 5 是示出实现系统虚拟机 510 的计算设备 500 中的逻辑层的分层架构图。计算机系统可以包括硬件组件(例如,执行硬件、内存、I/O 设备等)508 和包括应用程序模块 502、操作系统 504 和虚拟化模块 506 的软件组件。运行在虚拟化模块 506 之上的软件被称为“客户”软件,以及支持虚拟化模块的底层平台被称为“客户”硬件。

[0089] 和进程虚拟机不同,系统虚拟机 510 提供多个操作系统(被称为“客户操作系统”)

能够在其上共存的完整的环境。同样地,客户硬件平台可以被配置为同时支持多个隔离的客户操作系统环境。并发执行的操作之间的隔离为系统增加了安全级别。例如,如果一个客户操作系统上的安全性被破坏,或者如果一个客户操作系统遭受失败,则运行在其它客户系统上的软件不受破坏/失败的影响。主机硬件平台还简化了应用开发者的工作,因为应用软件不需要关心应用将最终在其上执行的计算设备的真实架构。

[0090] 可以使虚拟化软件模块 506 逻辑地位于主机硬件与客户软件之间。虚拟化软件可以运行在真实的硬件上(本机)或运行在操作系统之上(主机),并且典型地被称为“管理程序”或虚拟机监测器(VMM)。在本机配置下,虚拟机软件在可用的最高特权模式下运行在真实的硬件上,以及客户操作系统在降低的特权的情况下运行,以便虚拟化软件能够截获和模仿将正常地访问或操作硬件资源的所有客户操作系统的动作。在主机配置下,虚拟化软件运行在现有的主机操作系统之上,并且可以依靠主机操作系统来提供设备驱动程序和其它较低级别的服务。在任一种情况下,客户操作系统中的每一个客户操作系统(例如,操作系统 504)以它们与物理硬件 508 相通信的相同的方式与虚拟化软件模块 506 相通信,将虚拟化模块 506 和硬件 508 的组合看作单个虚拟机 510。这允许每一个客户操作系统(例如,操作系统 504)在专有地使用硬件 508 中的处理器、外围设备、I/O、MMU 和内存的错觉下操作。

[0091] 如上文关于图 3 讨论的,操作系统可以负责跨越多个进程划分物理内存。这可以由内存地址空间转换过程来实现。在内存地址空间转换过程中,操作系统向每一个应用程序分配虚拟地址,并且然后在程序的执行之前,基于虚拟地址分配物理地址。然而,在包括运行在虚拟机之上的客户操作系统的系统中,由客户操作系统分配的内存地址不是真实的物理地址,而是中间物理地址。在这样的系统中,通常由管理程序执行对真实物理内存的分配,可能需要所述管理程序来维护虚拟地址、中间物理地址与物理地址之间的关系。

[0092] 大部分处理器系统仅支持单级内存地址转换过程,并且需要管理程序来管理虚拟地址、中间物理地址与物理地址之间的关系。这通常通过管理程序维护其自己的转换表(被称为影子转换表)来实现,可以通过解译客户操作系统的转换表中的每一个转换表来导出该转换表。在这样的系统上,管理程序确保对客户操作系统的转换表的所有改变被反映在影子结构中,以及实施保护并将访问故障重定向到恰当的级。如上文讨论的,这些操作增加了管理程序的复杂度,并且对执行、维护和/或管理管理程序增加了显著的开销。

[0093] 和上文讨论的单级处理器不同,一些处理器系统(例如,ARM v7-A)为两级内存转换提供硬件协助。例如,ARM 处理器可以包括使得客户操作系统能够在第一级(即,第一级转换)中将虚拟地址转换到中间物理地址,以及针对硬件,在第二级(即,第二级转换)中将中间物理地址转换至物理地址的虚拟化扩展。这样的虚拟化扩展降低与执行、维护和/或管理管理程序相关联的开销,并且改善计算设备的性能。

[0094] 图 6 示出了与在实现系统虚拟机的计算设备 600 上按两级分配内存相关联的示例逻辑组件和地址转换。客户操作系统 610(例如,HL0S)的内存管理系统可以向应用程序/进程(A_0, A_n)中的每一个分配虚拟地址空间 602、604。例如,可以由虚拟内存管理器(例如,客户 OS VMM)来分配虚拟地址空间 602、604。每一个应用程序/进程(A_0, A_n)可以被分配其自己的虚拟地址空间 602、604,以及每一个虚拟地址空间 602、604 可以包括一个或多个虚拟地址 $VA_0 616$ 、 $VA_n 618$ 。

[0095] 在图 6 中示出的示例中,将内存地址转换成两级。在第一级转换 612 中,客户操作系统 610 的虚拟内存管理器(客户 OS VMM)可以将虚拟地址 VA_0 616、 VA_n 618 映射到中间物理地址空间 606 中的中间物理地址 IPA_0 626、 IPA_n 628。在第二级转换 614 中,管理程序和/或虚拟化扩展可以将中间物理地址 IPA_0 626、 IPA_n 628 映射到物理地址空间 608 中的物理地址 PA_0 636、 PA_n 638。第一级转换 612 可以独立于第二级转换 614 执行,以及在现有的系统中,执行第二级转换 614 的组件不基于内存的特性来分配物理地址。

[0096] 图 7 示出了当管理程序被禁用时,与在实现系统虚拟机的计算设备 700 上按两级分配内存相关联的示例逻辑组件和地址转换。在一个方面,当被禁用时,管理程序不能参加第二级转换 708,直到沙盒会话开始为止。例如,沙盒可以不是必须的,仅当客户操作系统(例如,HLOS)在计算设备上执行时才是必须的。因此,在各个方面,管理程序可以通过当确定不需要沙盒时不参加第二级转换来更有效地执行。

[0097] 在一个方面,当管理程序被禁用时,在第一级转换 706 中,如上文关于图 6 讨论的,HLOS 可以将虚拟地址空间 710 中的虚拟地址映射到 HLOS 的中间物理地址空间 720 中的中间物理地址。例如,HLOS 可以分别将虚拟地址 VA_0 712 和 VA_n 714 转换/映射到中间物理地址 IPA_0 722 和 IPA_n 724。在另一示例中,HLOS(或在 HLOS 上操作的 MMU)可以通过执行虚拟地址空间 710 与中间物理地址空间 720 之间的第一级转换 706 来分配虚拟内存块,供应用 A_0 至 A_n 使用。

[0098] 在又一方面,当被禁用时,管理程序不可以通过第二级转换 708 执行从中间物理地址空间 720 到物理地址空间 730 的转换。在这个方面,HLOS 可以绕过第二级转换 708。因此,因为允许 HLOS 绕过第二级转换 708,所以 HLOS 可以直接从物理地址空间 730 分配内存。通过使得 HLOS 能够绕过第二级转换 708,管理程序确保中间物理地址等于物理地址。因此,在示例中,中间物理地址空间 720 中的 IPA_n 724 等于物理地址空间 730 中的 PA_n 734。类似地,中间物理地址空间 720 中的 IPA_0 722 等于物理地址空间 730 中的 PA_0 732。

[0099] 图 8 示出了在沙盒会话期间,与在计算设备 800 上按两级分配内存相关联的示例逻辑组件和地址转换。在各个方面,响应于对沙盒会话开始的检测,可以启用管理程序。

[0100] 在一个方面,沙盒会话可以是在其中 HLOS 必须与受保护的内容相隔离的情况。受保护的内容可以包括安全应用、运行在计算设备上的第二操作系统或可能需要被单独地处理或存储的别的内容。例如,数字信号处理器(即,“DSP”)可以接收安全的视频信号(即,受保护的内容)用于处理。在本示例中,安全的视频信号可能需要与 HLOS 分开处理,以保持视频信号的完整性和/或安全性。在又一方面,响应于确定安全的视频信号需要被单独地处理,诸如 ARM 信任区®的安全性监测器可以警告管理程序沙盒会话已经开始。响应于对该警告的接收,管理程序可以被启用,并且可以重新开始实现从中间物理地址空间到物理地址空间的第二级转换。

[0101] 如图 8 中示出的,一旦沙盒会话已经开始,并且管理程序被启用,管理程序就可以重新开始第二级转换 808 和 846。在一个方面,与如上文关于图 6 描述的 HLOS 如何分配内存相类似,可以使用第一级转换 844 和第二级转换 846 二者来处理受保护的内容。例如,关于图 2 讨论的,安全环境(例如,在安全的虚拟机内操作的 DSP)可以通过至安全网络的连接接收安全的视频信号。在本示例中,DSP 可以从虚拟地址空间 850 向运行在 DSP 上的视频处理应用分配一个或多个 4kb 的内存块(例如, VA_{cp} 852),用于存储接收的安全的视频

信号。DSP 还可以通过执行第一级转换 844 来维护从虚拟地址空间 850 中的 VA_{cp} 852 至中间物理地址空间 860 中的中间物理地址 IPA_{cp} 862 的映射。

[0102] 在沙盒会话期间（即，在从中间物理地址空间 860 的安全环境的内存分配期间），HLOS 还可以执行内存分配。然而，因为管理程序被启用，并且已经重新开始第二级转换 808，所以 HLOS 不再具有不受限制的直接从整个物理地址空间 830 分配内存的能力。

[0103] 因此，在一个方面，管理程序可以限制物理地址空间 830 中的物理地址可用于 HLOS 的分配。换句话说，HLOS 仍然可以执行直接至物理地址空间 830 的内存分配，但是，管理程序可以限制 HLOS 访问物理地址空间 830 的一些部分的能力。例如，HLOS 可以分配虚拟内存 VA_0 712 和 VA_n 714，其在第一级转换 706 之后，分别映射到中间物理地址空间 720 中的 IPA_0 722 和 IPA_n 724。此外，如图 7 中示出的，仍然可以将 IPA_0 722 和 IPA_n 724 映射到 PA_0 732 和 PA_n 734，但是，当管理程序被禁用时（即，当 HLOS 能够绕过第二级转换 808 时），然而 HLOS 如关于图 7 描述的，能从整个物理地址空间 830 分配内存，当管理程序被启用时，HLOS 可以使用较小的物理地址集合。

[0104] 当管理程序被启用并且执行第二级转换 808、846 时，管理程序可以从以前可用于 HLOS 的物理地址向沙盒化的组件分配内存。例如，管理程序可以将 IPA_{cp} 862 映射到现在不再可用于 HLOS 的 PA_{cp} 872。因此，当被启用时，管理程序可以通过向例如沙盒化的实体分配内存来对当管理程序被禁用时可用于 HLOS 的物理地址“穿孔”。在执行从 HLOS 的中间物理地址空间 720 向物理地址空间 830 的第二级转换 808 中，管理程序可以隐藏物理地址空间 830 中“被穿孔的”物理地址，由此防止 HLOS 访问“被穿孔的”物理地址。因此，例如，在管理程序向沙盒化的组件分配 PA_{cp} 872 之后，HLOS 可能不再使用那个物理地址。

[0105] 通过停止允许 HLOS 绕过级 2 转换 808，管理程序可以管理 HLOS（并且因此沙盒化的组件）最终使用的物理地址。因此，当存在更多个必须被隔离的应用、进程、操作系统等时，管理程序可以通过重新开始除了别的以外的级 2 转换（即，通过直接管理对物理内存的访问）来创立沙盒。

[0106] 图 9 示出了在沙盒会话和共享虚拟内存进程期间，与在计算设备 900 上按照两级分配内存相关联的示例逻辑组件和地址转换。在各个方面，响应于和共享实体（例如，DSP）一起检测沙盒会话的开始，管理程序可以被启用，并且 HLOS 和共享实体可以共享物理地址空间 930 中的一些物理地址。

[0107] 在下文关于图 11 和 12 描述的方面，计算设备可以诸如在 HLOS 与数字信号处理器（DSP）之间发起共享虚拟内存会话。在又一方面，HLOS 与 DSP 之间的共享虚拟内存会话可以包括配置 HLOS 和 DSP 来共享对物理地址空间 930 中的物理地址集合的访问。例如，在 HLOS 和 DSP 需要共享数据结构、例程等时，HLOS 和 DSP 可以经历共享虚拟内存会话。通过共享直接内存访问，HLOS 和 DSP 可以在不必须复制和传输被存储在物理地址空间 930 中的信息的情况下有效地共享信息。

[0108] 如图 9 中示出的，HLOS 和 DSP 可能已经分别向操作在 HLOS 和 DSP 上的应用、进程等分配了相应虚拟地址空间 910、950 中的虚拟内存（即， VA_{SVMl} 914 和 VA_{SVMn} 912）。例如，操作在 HLOS 和 DSP 上的应用可以共享某些数据结构、函数或库。HLOS 和 DSP 可以执行第一级转换 906、944，以分别将虚拟地址 VA_{SVMl} 914 和 VA_{SVMn} 912 映射到共享实体的中间物理地址空间 960 和 HLOS 的中间物理地址空间 920 中的每一个中的 IPA_{SVMl} 924 和 IPA_{SVMn} 922。

[0109] 在另一方面,因为响应于沙盒会话的开始,管理程序被启用,所以管理程序可以激活将来自 HLOS 的中间物理地址空间 920 的中间物理地址映射到物理地址空间 930 的第二级转换 908。管理程序还可以激活将共享实体的中间物理地址空间 960 中的中间物理地址映射到物理地址空间 930 的第二级转换 946。

[0110] 在一个方面,如关于图 8 描述的,通过使能从 HLOS 的中间物理地址空间 920 到物理地址空间 930 的第二级转换 908,管理程序可以限制 HLOS 使用的物理地址空间 930 中的物理地址(即,移除至一些物理地址的映射,以便 HLOS 不可以访问物理地址空间 930 中的那些物理地址)。如图 9 中示出的,管理程序可以维护从 IPA_{HLOS} 926、 IPA_{SVM1} 924 和 IPA_{SVMn} 922 分别至物理地址 PA_{HLOS} 936、 PA_{SVM1} 934 和 PA_{SVMn} 932 的映射,以便从 HLOS 的中间物理地址空间 920 到物理地址空间 930 的映射 940 确保中间物理地址等于物理地址。类似地,管理程序可以维护除了别的之外的从共享实体的中间物理地址空间 960 中的 IPA_{SVM1} 924 和 IPA_{SVMn} 922 分别至物理地址空间 930 中的 PA_{SVM1} 934 和 PA_{SVMn} 932 的共享映射 941。

[0111] 在另一方面,管理程序可以对共享实体实现“部分”沙盒。在这个方面,管理程序可以移除从 HLOS 到被分配给共享实体、不与 HLOS 共享的物理地址(例如, $PA_{non-SVM}$ 962)的映射。共享实体和 HLOS 均可以维护至不被共享的物理地址空间中的内存的映射。例如, DSP 可以维护与可以是例如 DSP 的内核的 $IPA_{non-SVM}$ 923 相关联的信息。在另一示例中, HLOS 可以维护被映射到不与共享实体共享的 PA_{HLOS} 936 的 HLOS 的中间物理地址空间 920 中的 IPA_{HLOS} 926 处的内存。

[0112] 然而,管理程序不可以移除从 HLOS 到被分配给共享实体但不与 HLOS 共享的物理地址的映射(例如, IPA_{SVM1} 924 和 IPA_{SVMn} 922)。通过不移除该映射,管理程序可以允许 HLOS 和共享实体来共享被存储在这些物理地址处的信息,诸如,至数据结构的指针、库、例程等。

[0113] 因此,通过管理从第二级转换 908 被移除的、从 HLOS 的中间物理地址空间 920 至物理地址空间 930 的映射,管理程序可以使得 HLOS 和共享实体能够共享被存储在某些物理地址处的信息,并且仍然可以实施对未被共享的地址(例如, $PA_{non-SVM}$ 962 和 PA_{HLOS} 936)的访问控制。

[0114] 图 10 示出了可以在用于沙盒会话期间选择性地启用管理程序的计算设备处理器(例如, CPU)中实现的一个方面的方法 1000。在框 1002 中,计算设备处理器可以初始化管理程序、安全性监测器和 HLOS。在一个方面,计算设备处理器可以通过使用 Linaro ARM8 安全引导流程和 Xen 方式授权表来引导到管理程序、安全性监测器和 HLOS 中来初始化管理程序、安全性监测器和 HLOS。在另一方面,安全性监测器可以是 ARM 信任区[®]。

[0115] 在又一方面,在初始化期间,可以由安全性监测器来对管理程序的代码和数据进行认证和/或签名。在初始化期间,还可以配置管理程序,以便其代码和数据对于外部的处理器(诸如数字信号处理器(DSP)或被包括在 DSP 中的 CPU)是不可访问的。在另一方面,当管理程序被启用时,对管理程序的认证和/或防止外部处理器访问管理程序的代码和数据可以确保未来的沙盒会话是安全的。

[0116] 在可选的确定框 1004 中,计算设备处理器可以确定是否存在与共享虚拟内存的并发异构计算会话。在一个方面, HLOS 和共享实体(例如, DSP)可以共享复杂的、包含指针的数据结构。如果计算设备处理器确定与共享虚拟内存的并发异构计算会话的情形存在的(即,确定框 1004 = “是”),则在框 1006 中,处理器可以与共享虚拟内存建立并发异构

计算会话。在一个方面,例如,HLOS 和 DSP 可以被配置为共享同一个第一级页表。下文关于图 11 和 12 详细地讨论与共享虚拟内存建立并发异构计算会话。计算设备处理器可以继续框 1008 中的操作。如果计算设备处理器确定不存在于共享虚拟内存的并发异构计算会话(即,可选的确定框 1004 = “否”),则处理器还可以继续框 1008 中的操作。

[0117] 在框 1008 中,计算设备处理器可以禁用管理程序。在一个方面,管理程序的默认状态可以是被禁用的。在另一方面,禁用管理程序可以禁用从中间物理地址空间至物理内存地址空间的第二级转换。禁用管理程序的其它结果可以包括停止限制 HLOS 对硬件中断、硬件定时器和输入/输出的访问。下文关于图 13 进一步讨论对管理程序的禁用。

[0118] 在框 1009 中,计算设备处理器可以监测管理程序上接收的信号,以开始沙盒会话。在一个方面,安全性监测器(例如,ARM 信任区®)可以接收或检测受保护的内容,并且向管理程序发送唤醒信号,以开始沙盒会话。例如,在安全的虚拟环境内操作的 DSP 可以接收安全的视频信号用于安全的处理。在本示例中,例如,DSP 可以被配置为将视频信号存储到对 HLOS 不可访问的物理内存地址空间的一部分中。

[0119] 在确定框 1010 中,计算设备处理器可以确定管理程序是否接收到信号以开始沙盒会话。如果计算设备处理器确定管理程序尚未接收到信号来开始沙盒会话(即,确定框 1010 = “否”),则处理器可以继续框 1009 中的操作。在一个方面,计算设备处理器可以继续监测供管理程序开始沙盒会话的信号。

[0120] 如果计算设备处理器确定管理程序确实接收到了用于开始沙盒会话的信号(即,确定框 1010 = “是”),则处理器可以在框 1012 中启用管理程序。在一个方面,启用管理程序可以包括重新开始第二级转换。下文关于图 14 和 15 进一步详细地讨论对管理程序的启用。

[0121] 然后,管理程序可以在框 1014 中实现访问控制。在一个方面,管理程序可以通过执行从中间物理地址至物理地址的第二级转换来实现访问控制。在又一方面,管理程序另外可以通过重新开始限制对 I/O、硬件中断和硬件定时器的访问来实现访问控制。下文关于图 16A 和 16B 进一步详细地描述实现访问控制的过程。

[0122] 在确定框 1016 中,管理程序可以确定沙盒会话是否结束。例如,沙盒会话可以在不存在必须被保护或与 HLOS 或其它进程、应用或组件隔离的内容时结束。例如,在上文给出的示例中,在 DSP 接收安全的视频信号时发起的沙盒会话可以在 DSP 已经处理安全的视频信号并且不再需要将视频信号的视频缓冲器存储到物理内存中之后结束。在另一方面,安全的虚拟环境中的安全性监测器或另一组件可以用信号通知管理程序沙盒会话结束。

[0123] 如果沙盒会话未结束(即,确定框 1016 = “否”),则管理程序可以继续执行框 1014 中的操作。否则(即,确定框 1016 = “是”),管理程序可以在框 1018 中拆除沙盒会话。在一个方面,作为执行沙盒会话拆除过程的结果,管理程序可以向计算设备的 HLOS 和各个其它组件返回“默认”状态或配置。下文关于图 17 进一步详细地描述沙盒会话的拆除。计算设备可以继续执行框 1008 中的操作。

[0124] 图 11 示出了用于发起并发异构计算会话的计算设备的各个组件之间的一个方面的信号和呼叫流程。在一个方面,在操作 1112 中,HLOS 1102 可以创建第一级转换表。HLOS 1102 还可以在操作 1114 中分配 HLOS 的虚拟机标识符(即,“HLOS_VMID”)中特定于应用的标识符(ASID)。在另一方面,管理程序 1104 可以向数字信号处理器(DSP)的第二级系

统内存管理单元 (SMMU) 1106 发送信号 1116, 以使用 HLOS 的 HLOS_VMID 来创建第二级转换表。在一个方面, DSP 第二级 SMMU 可以使用第二级转换表来执行从 HLOS 的中间物理地址空间到物理地址空间的第二级转换。

[0125] 在一个方面, HLOS 可以向 DSP 的管理程序 1108 发送信号 1118, 请求使用 HLOS 的 HLOS_VMID、选择的特定于应用的标识符和第一级转换表来创建共享虚拟内存 (SVM) 进程。DSP 的管理程序 1108 可以利用信号 1120 来发起 DSP 的共享虚拟内存进程 1110, 其针对 DSP 的内存管理单元 (MMU) 来对第一级转换表编程, 以及然后发起 DSP 的共享虚拟内存进程 1110。

[0126] 图 12 示出了可以在用于在 HLOS 与 DSP 之间发起共享虚拟内存进程的计算设备中实现的一个方面的方法 1006a。计算设备处理器可以在处理器确定存在与共享虚拟内存的并发异构计算会话时 (即, 确定框 1004 = “是”), 开始方法 1006a。在框 1204 中, 计算设备可以配置 HLOS 来创建第一级转换表。在一个方面, HLOS 可以使用第一级转换表来将虚拟地址映射到中间物理地址。计算设备还可以在框 1206 中配置 HLOS 来分配 HLOS 的虚拟机标识符 (即, HLOS_VMID) 中特定于应用的标识符。

[0127] 在框 1208 中, 计算设备可以配置管理程序向 DSP 第二级 SMMU 发送基于 HLOS_VMID 的第二级配置。在一个方面, DSP 第二级 SMMU 可以基于 HLOS_VMID 针对 HLOS 创建第二级转换表。SMMU (或管理程序) 可以使用第二级转换表来执行从 HLOS 的中间物理地址空间到计算设备的物理内存地址空间的第二级转换。

[0128] 在框 1210 中, 计算设备可以配置 HLOS 来请求 DSP 的管理程序使用 HLOS_VMID、选择的 ASID 和 HLOS 第一级转换表来创建共享虚拟内存进程。

[0129] 计算设备还可以在框 1212 中配置 DSP 的管理程序来对第一级内存管理单元 (MMU) 编程, 并发起共享虚拟内存进程。在一个方面, DSP 的第一级 MMU 可以发起与 HLOS 的第一级转换表相同的第一级转换表。因此, 在本方面, 因为 HLOS 和 DSP 共享相同的第一级转换表, 所以它们可以共享虚拟内存。

[0130] 随着共享虚拟内存进程在框 1212 中的完成, 在不存在针对沙盒会话的需求时, 计算设备处理器可以如上文关于图 10 描述的在框 1008 中禁用管理程序。

[0131] 图 13 示出了用于禁用计算设备上的管理程序的一个方面的方法 1008a。

[0132] 在框 1304 中, 管理程序可以配置所有 SMMU 第一级上下文库来绕过第二级转换。管理程序还可以在框 1306 中关闭 HLOS 的第二级转换。

[0133] 在一些方面, 管理程序在被禁用时可以暂停各个其它活动。例如, 管理程序可以在可选框 1308 中可选地暂停限制 I/O 访问。管理程序还可以在可选框 1310 中暂停限制硬件中断访问。此外, 在可选框 1312 中, 管理程序可以暂停限制硬件定时器访问。

[0134] 管理程序可以在如上文关于图 10 描述的决定框 1010 中确定管理程序是否接收用于开始沙盒会话的信号。

[0135] 在一些方面, 各个管理程序功能 (例如, 访问控制、沙盒内存等) 可以跨越集成电路边界和 / 或芯片边界被禁用。在一个方面, 在包括分立的芯片组 (例如, 调制解调器芯片和应用处理器芯片) 的像融合的芯片组的组合中, 主芯片组 (即, 主管理程序) 可以在管理程序的功能已经被禁用时暂停其它芯片组中的沙盒内存或其它访问控制。例如, 应用处理器芯片组中的主管理程序可以暂停从调制解调器或 DSP 芯片组中的中间物理地址至物理

地址的第二级转换。因此,当管理程序的功能被禁用时,主管理程序可以暂停多个分立的芯片中的这些功能。

[0136] 图 14 示出了当针对内容受保护的信号建立沙盒会话时,在计算设备上操作的多个组件之间的呼叫流程 1400。在各个方面,针对视频缓冲器的页面可以是 4kb 页面,并且可以使 HLOS 的第二级转换页表成碎片。

[0137] 在一个方面,可以将信号 1402 发送至诸如安卓 MM 框架 1450、OpenMax (OMX) 组件 1452、V4L2 视频驱动器 1454、内核页面分配器 1456、管理程序 1458、SMMU 1460 和核 1462 的各个组件,以初始化具有安全认证的固件。在一个方面,安全认证可以是 ARM 信任区[®]。在另一方面,如果视频在 DSP 上,则计算设备可以加载 DSP 视频应用和编解码器。

[0138] 在一个方面,安卓 MM 框架 1450 可以向 OMX 组件发送信号 1404 来初始化后者。OMX 组件 1452 可以发送信号 1406 以在 V4L2 视频驱动器 1454 上设置编解码器。OMX 组件 1452 还可以向 V4L2 视频驱动器 1454 发送缓冲器大小查询信号 1408。安卓 MM 框架 1450 还可以向 OMX 组件 1452 发送空载信号 1410。

[0139] 在另一方面,OMX 组件 1452 可以向 V4L2 视频驱动器 1454 发送流开启输入信号 1412。V4L2 视频驱动器 1454 可以向核 1462 发送主机固件接口 (HFI) 会话初始化信号 1414。核 1462 可以通过向 V4L2 视频驱动器 1454 发送“HFI 会话完成”来响应 HFI 会话初始化信号 1414。

[0140] 安卓 MM 框架可以向 OMX 组件 1452 发送内存分配信号 1418 (即,“ioctl ION_IOC_ALLOC”)。在另一方面,安卓 MM 框架可以发送内存分配信号 1418 来管理页面池。OMX V4L2 视频驱动器 1454 可以向内核页面分配器 1456 发送“ion_alloc (ION 堆)”信号 1420。

[0141] 然后,内核页面分配器 1456 可以向管理程序 1458 发送 VMM_CALL 信号 1422,以移除处理器第二级映射。在一个方面,这个信号 1422 可以通知管理程序沙盒会话已经开始,并且必须从对 HLOS 可访问的那些物理地址中移除某些物理内存位置。在又一方面,管理程序 1458 可以向 SMMU 1460 发送第二级转换映射信号 1424。在一个方面,SMMU 可以实现这些第二级转换 (即,SMMU 可以重新开始第二级转换)。在又一方面,SMMU 可以维护从 HLOS 至可访问的物理地址的映射 (即,映射至尚未对 HLOS 隐藏的物理地址)。在又一方面,如果视频在 DSP 上,则内核页面分配器 1456 可以另外用信号通知管理程序 1458 来将页面映射至 DSP 第二级转换映射。

[0142] 在另一方面,V4L2 视频驱动器 1454 可以向核 1462 发送主机固件接口设置缓冲器 1426,并且可以向安卓 MM 框架发送指示所有缓冲器就绪的空闲信号 1428。然后,安卓 MM 框架 1450 可以用信号通知 1430 OMX 组件来转换以执行。然后,安卓 MM 框架 1450 还可以向 OMX 组件 1452 发送信号 1432 来利用报头对第一缓冲器排队。OMX 组件 1452 还可以向 V4L2 视频驱动器 1454 来发送信号 1434,以利于报头对第一缓冲器排队。

[0143] V4L2 视频驱动器 1454 可以向内核页面分配器 1456 发送将第一缓冲器映射到核 1462 的信号 1436。内核页面分配器 1456 可以向 SMMU 1460 发送信号 1438 来将第一缓冲器映射到第一级转换输入上下文。V4L2 视频驱动器 1454 还可以向核 1462 发送利用报头对第一缓冲器排队的信号 1440。

[0144] 图 15 示出了可以在用于启用管理程序的计算设备处理器中实现的一个方面的方法 1012a (即,在管理程序在上文关于图 10 描述的框 1010 中被禁用之后)。在框 1504 中,

计算设备可以通过将 HCR.VM 设置为“1”来启用 PL0 和 PL1 (即, 权限级别 1 和权限级别 2) 第二级 MMU。在框 1506 中, 计算设备可以配置在管理程序模式下采取中断请求 (“IRQ”)。在一个方面, 计算设备可以通过将 SCR.IMO 设置为“1”来完成这个。

[0145] 计算设备还可以在框 1508 中调用 SMMU 驱动器, 以致使所有活动的 SMMU 上下文处于在其中利用第二级转换嵌套第一级转换的状态。计算设备可以通过将 SMMU_CBArn.type 元素设置为“0b11”来设置这个状态。

[0146] 在一个方面, 一旦这些步骤被完成, 管理程序就可以在如上文关于图 10 描述的框 1012 中被启用。在框 1012 中被启用之后, 管理程序可以在可选框 1510 中可选地与 DSP 开始处理器间通信 (IPC)。管理程序还可以在可选框 1512 中可选地处理 SMMU 故障。

[0147] 在一些方面, 管理程序可以在被启用时重新开始各个其它活动。例如, 管理程序可以在可选框 1514 中重新开始限制 I/O 访问。管理程序还可以在框 1516 中重新开始限制对硬件中断访问。在框 1518 中, 管理程序另外可以重新开始限制硬件定时器访问。

[0148] 然后, 计算设备处理器可以在如上文关于图 10 描述的框 1014 中实现访问控制。

[0149] 在一些方面, 各个管理程序的功能 (例如, 访问控制、沙盒内存等) 可以跨越集成电路边界和 / 或芯片边界被启用。如上文关于图 13 讨论的, 在一个方面, 主芯片组 (即, 主管理程序) 可以通过例如外部部件互连快速接口来控制其它芯片组中的沙盒内存。例如, 应用处理器芯片组中的主管理程序可以控制调制解调器或 DSP 芯片组中从中间物理地址到物理地址的转换。因此, 在管理程序的功能被启用时, 主管理程序可以跨越多个分立的芯片来执行那些功能。

[0150] 图 16A 和 16B 示出了可以在用于在沙盒会话期间执行第二级转换的计算设备处理器上实现的一个方面的方法。在各个方面, 管理程序可以管理在其中各个组件 (例如, HLOS 或 DSP) 可以访问的物理内存地址空间中的位置。

[0151] 图 16A 示出了用于在沙盒会话期间管理程序分配内存的一个方面的方法 1014a。当管理程序在如上文关于图 10 描述的框 1012 中被启用时。

[0152] 管理程序可以在可选的确定框 1604 中确定当前是否存在共享虚拟内存状况。在一个方面, 共享虚拟内存状况可以在例如 HLOS 在与另一组件共享虚拟内存时存在。如果存在共享虚拟内存状况发生 (即, 确定框 1604 = “是”), 则管理程序可以执行下文关于图 16B 描述的方法 1014b。否则 (即, 确定框 1604 = “否”), 管理程序可以在框 1608 中监测分配内存的尝试。管理程序可以在确定框 1610 中确定 HLOS 是否在尝试分配内存。在一个方面, HLOS 可以尝试为当前操作在 HLOS 上的应用或进程分配内存。例如, HLOS 可以通过创建由应用访问的虚拟地址空间来为那个应用分配内存。如果处理器确定 HLOS 在尝试分配内存 (即, 确定框 1610 = “是”), 则管理程序可以在框 1612 中从对 HLOS 可访问的物理内存地址空间向 HLOS 提供物理地址。在一个方面, HLOS 不可以任意地使用该物理地址空间, 因为该物理地址中的一些地址从 HLOS 的第二级转换映射中被移除, 并且可以被分配给受保护的内容。例如, HLOS 不可以具有至被分配给 DSP 用于存储 4kb 视频缓冲器的物理地址的映射。然后, 管理程序可以在上文关于图 10 描述的确定框 1016 中确定沙盒会话是否结束。

[0153] 如果 HLOS 没在尝试分配内存 (即, 确定框 1610 = “否”), 则管理程序可以在确定框 1611 中确定沙盒化的组合是否在尝试分配内存。例如, 管理程序可以确定 DSP 处理安

全的视频信号是否在尝试将 4kb 的视频缓冲器存储到物理内存中。如果沙盒化的组件没在尝试分配内存（即，确定框 1611 = “否”），则管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0154] 如果管理程序确定沙盒化的组件在尝试分配内存（即，确定框 1611 = “是”），则管理程序可以在框 1614 中从物理地址空间中对 HLOS 可访问的物理地址中移除将被提供给沙盒化的组件的物理地址。管理程序还可以在框 1616 中向沙盒化的组件提供物理地址，所述物理地址是来自物理地址空间中的可用的物理地址。在一个方面，可用的物理地址可以是物理地址空间中尚未被分配给 HLOS 的物理地址。换句话说，可用的物理地址是“空闲”的内存地址。在一个方面，一旦管理程序分配物理地址空间中的内存供沙盒化的组件使用，HLOS 就可以在沙盒会话期间不再使用那个物理内存。例如，一旦用于安全的视频的 4kb 视频缓冲器被存储在特定的物理地址处，HLOS 就可以不再具有至那个物理地址的映射（即，HLOS 可能不再能够“看到”那个物理地址来分配它）。然后，管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0155] 图 16B 示出了当 HLOS 在与另一组件共享虚拟内存时（即，当确定框 1604 = “是”时），可以在用于在沙盒会话期间分配内存的管理程序中实现的一个方面的方法 1014b。在确定框 1624 中，管理程序可以确定 HLOS 是否在尝试分配物理地址。例如，HLOS 可以在尝试分配物理地址空间中的某些物理地址供运行在 HLOS 上的应用使用。如果管理程序确定 HLOS 在尝试分配物理地址（即，确定框 1624 = “是”），则管理程序可以在框 1612 中向 HLOS 提供物理地址，所述物理地址是来自物理内存地址空间中对 HLOS 可访问的物理地址。在上文关于图 16A 描述的方面，例如，在管理程序允许 HLOS 分配那些物理地址时，对于 HLOS，物理地址可能是可访问的。换句话说，对于 HLOS 可访问的物理地址尚未被分配给沙盒化的组件并且对 HLOS 隐藏。在向 HLOS 分配物理地址之后，管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0156] 如果管理程序确定 HLOS 没在尝试分配物理地址（即，确定框 1624 = “否”），则管理程序可以在确定框 1626 中确定沙盒化的组件是否在尝试分配物理地址。例如，在沙盒化的组件内操作的 DSP 可以尝试访问某个物理地址，以存储 4kb 的视频缓冲器。如果管理程序确定沙盒化的组件在尝试分配物理地址（即，确定框 1626 = “是”），则管理程序可以在框 1628 中从物理内存地址空间中对 HLOS 可访问的 PA 中移除将被提供给沙盒化的组件的物理地址。在一个方面，可以将被分配给沙盒化的组件的物理地址对 HLOS 隐藏。换句话说，管理程序可以移除从 HLOS 至被分配给沙盒化的组件的那些物理地址的第二级映射。管理程序还可以在框 1632 中向沙盒化的组件提供物理地址，所述物理地址是来自物理地址空间中的可用的物理地址。在一个方面，可用的物理地址可以包括管理程序尚未留出供 HLOS 使用的那些物理地址（即，物理地址空间中“空闲”的物理地址）。在向沙盒化的组件分配物理地址之后，管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0157] 如果管理程序确定沙盒化的组件没在尝试分配物理地址（即，确定框 1626 = “否”），则管理程序可以在确定框 1630 中确定共享实体是否在尝试分配物理地址。在一个方面，共享实体可以是操作在计算设备上的、与 HLOS 共享虚拟内存的组件。在另一方面，HLOS 和另一实体可以通过共享至物理地址的指针来共享虚拟内存。换句话说，HLOS 和共享实体可能能够访问或分配物理内存地址空间中相同的物理地址。

[0158] 如果管理程序确定共享实体没在尝试分配物理地址（即，确定框 1630 = “否”），则管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。否则（即，确定框 1630 = “是”），管理程序可以在确定框 1632 中确定共享实体是否在尝试分配共享物理地址。换句话说，管理程序可以确定共享实体是否在尝试利用、改变、访问、分配或以别的方式对与 HLOS 共享的物理地址读取或写入。

[0159] 如果管理程序确定共享实体在尝试分配共享物理地址（即，确定框 1632 = “是”），则管理程序可以在框 1636 中向共享实体提供共享物理地址。在一个方面，HLOS 还可以访问并分配共享物理地址。换句话说，管理程序不可以向 HLOS 隐藏被分配给共享实体的共享物理地址。管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0160] 否则，如果管理程序确定共享实体没在尝试分配共享物理地址（即，确定框 1632 = “否”），则管理程序可以在框 1634 中从物理内存地址空间中对 HLOS 可访问的物理地址中移除将被提供给共享实体的物理地址。在框 1638 中，管理程序可以向共享实体提供物理地址，所述物理地址是来自物理地址空间中的可用的物理地址。在一个方面，在共享实体没在访问与 HLOS 共享的物理地址时，管理程序可以像如上文描述的沙盒化的组件来处理共享实体。在分配物理地址之后，管理程序可以在上文关于图 10 描述的决定框 1016 中确定沙盒会话是否结束。

[0161] 图 17 示出了可以在用于执行会话拆除的管理程序中实现的一个方面的方法 1018a。当管理程序确定沙盒会话结束时（即，确定框 1016 = “是”），管理程序可以在框 1704 中释放所有的该沙盒化的组件的缓冲器。例如，管理程序可以释放被存储在物理内存地址空间中各个物理地址中的 4kb 的视频缓冲器。在一个方面，通过释放这些缓冲器，管理程序可以准备这些物理地址要对 HLOS 可访问。

[0162] 在框 1706 中，管理程序可以恢复第二级转换页表，以移除所有存储残片。在一个方面，管理程序可以恢复物理地址空间中已经被分配给沙盒化的组件的内存穿孔的物理地址。在另一方面，管理程序可以添加第二级映射，所述第二级映射可以使得 HLOS 能够访问在向沙盒化的组件分配那些物理地址之后管理程序隐藏的物理地址。管理程序还可以在如上文关于图 10 描述的框 1008 中被禁用。因此，在一个方面，在执行会话拆除过程之后，管理程序可以致使 HLOS 回到在其中 HLOS 可以直接从整个物理内存地址空间分配内存以及然后被禁用的状况。

[0163] 适合于与各个方面一起使用的典型的计算设备 1800 将具有图 18 中示出的共同的组件。例如，典型的计算设备 1800 可以包括被耦合至内部存储器 1801、显示器 1803 以及至扬声器 1864 的处理器 1802。此外，计算设备可以具有被耦合至 1802 的用于发送和接收电磁辐射的天线 1804。在一些方面，计算设备 1800 可以包括一个或多个专用的或通用的处理器 1805、1824，它们可以包括片上系统。计算设备典型地还包括用于接收用户输入的小键盘或微型键盘（未示出）和菜单选择按钮 1808a、1808b。计算设备还可以包括用于开启或关闭计算设备的电源按钮 1834。

[0164] 诸如图 19 中示出的膝上型电脑 1900 的其它形式的计算设备还可以实现各个方面，并且从各个方面受益。向膝上型电脑 1900 的计算设备典型地包括被耦合至内部存储器 1901 和诸如硬盘驱动器 1905 或闪存存储器的大容量非易失性存储器以及显示器 1909 的处理器 1902。计算设备还可以包括用于接收用户输入的键盘 1908 和选择按钮 1907。

[0165] 在实现各个方面的计算设备中使用的处理器 1802、1805、1824、1902 可以是能够由处理器可执行软件指令（应用）配置的以执行包括本文描述的各个方面的功能的各种各样的功能的任意可编程微处理器、微型计算机或多处理器芯片。典型地，可以在软件应用和处理器可执行指令被访问并加载到处理器 1802、1805、1824、1902 中之前，可以将它们存储到内部存储器 1801、1901 中。在一些计算设备中，处理器 1802、1805、1824、1902 可以包括足以存储应用软件指令的内部存储器。

[0166] 在一些计算设备中，安全的内存可以是耦合至处理器 1802、1805、1824、1902 的单独的内存芯片。在许多计算设备中，内部存储器 1801、1901 可以是易失性存储器或诸如闪存存储器的非易失性存储器或这二者的混合。存储器可以包括任意数量的不同类型的存储技术，包括现有技术中已知的相变存储器（PCM）、动态随机存取存储器（DRAM）、静态随机存取存储器（SRAM）、非易失性随机存取存储器（NVRAM）、伪静态随机存取存储器（PSRAM）、双数据速率同步动态随机存取存储器（DDR SDRAM）以及其它随机存取存储器（RAM）和只读存储器（ROM）技术。出于本描述的目的，对存储器的一般提及指代所有由处理器 1802、1805、1824、1902 可访问的存储器，包括内部存储器、被插入到计算设备中的可移动存储器以及处理器内的存储器。

[0167] 提供前述方法描述和过程流程图，仅作为说明性的示例，并不旨在要求或暗示各个方面的步骤必须按照呈现的顺序执行。如将被本领域技术人员意识到的是，可以按照任意顺序执行前述方面中步骤的顺序。诸如“此后”、“然后”、“接下来”等词不旨在限定步骤的顺序；这些词仅仅被用来通过方法的描述来引导读者。此外，以例如使用冠词“一（a）”、“一个（an）”或“所述”的单数形式对权利要求要素的任意提及将不被解释为将该要素限定为单数。

[0168] 可以将结合本文公开的方面描述的各个说明性的逻辑框、模块、电路和算法步骤实现为电子硬件、计算机软件或这二者的组合。为了清楚地示出硬件和软件的这个可互换性，上文已经根据它们的功能概括地描述了各个说明性的组件、框、模块、电路和步骤。这样的功能是被实现为硬件还是软件取决于被施加到整个系统上的特定应用和设计约束。技术人员可以针对每一个特定应用以变通的方式实现描述的功能，但是，这样的实现方式的决定不应当被解释为导致对本发明的范围的脱离。

[0169] 可以利用被设计为执行本文描述的功能的通用处理器、数字信号处理器（DSP）、多媒体广播接收机芯片内的 DSP、专用集成电路（ASIC）、现场可编程门阵列（FPGA）或其它可编程逻辑器件、分立的门或晶体管逻辑、分立的硬件组件或它们的任意组合，来实现或执行结合本文公开的方面描述的被用于实现各个说明性的逻辑、逻辑框、模块和电路的硬件。通用处理器可以是微处理器，但是，在替代方案中，该处理器可以是任意传统的处理器、控制器、微控制器或状态机。处理器还可以被实现为例如 DSP 和微处理器的组合的计算设备的组合、多个微处理器、结合 DSP 内核的一个或多个微处理器、或任意其它这样的配置。替代地，可以由针对给定功能的电路来执行一些步骤或方法。

[0170] 在一个或多个示例性方面，可以在硬件、软件、固件或它们的任意组合中实现描述的功能。如果在软件中实现，则可以将该功能作为一个或多个指令或代码被存储在非暂时性计算机可读介质上或非暂时性处理器可读介质上。可以将本文公开的方法或算法的步骤体现在其可以存在于非暂时性处理器可读存储介质或非暂时性计算机可读存储介质上的

处理器可执行软件模块中。非暂时性计算机可读或处理器可读存储介质可以是能够由计算机或处理器访问的任意存储介质。通过示例而非限定方式,这样的非暂时性计算机可读或处理器可读介质可以包括 RAM、ROM、EEPROM、闪速存储器、CD-ROM 或其它光盘存储、磁盘存储、或其它磁性存储设备、或能够被用于以指令或数据结构的形式存储期望的程序代码并且可以被计算机存取的任意其它介质。如本文使用的,磁盘和光盘包括压缩光盘 (CD)、激光盘、光盘、数字多功能光盘 (DVD)、软盘和蓝光盘,其中,磁盘通常磁性地复制数据,而光盘则利用激光光学地复制数据。也将上文的组合包括在非暂时性计算机可读和处理器可读介质的范围内。此外,方法或算法的操作可以作为一个、任意组合或一组代码和 / 或指令存在于可以被并入到计算机程序产品中的非暂时性处理器可读介质和 / 或计算机可读介质上。

[0171] 提供了公开的方面的先前描述,使得本领域技术人员能够实施或使用本发明。对于本领域技术人员来说,对这些方面的各种修改将是显而易见的,并且在不脱离本方面的精神或范围的情况下,可以将本文定义的一般原则应用于其它方面。因此,本发明不是要受限于本文示出的方面,而是要符合与本文公开的下面的权利要求和原则以及新颖的特征相一致的最广的范围。

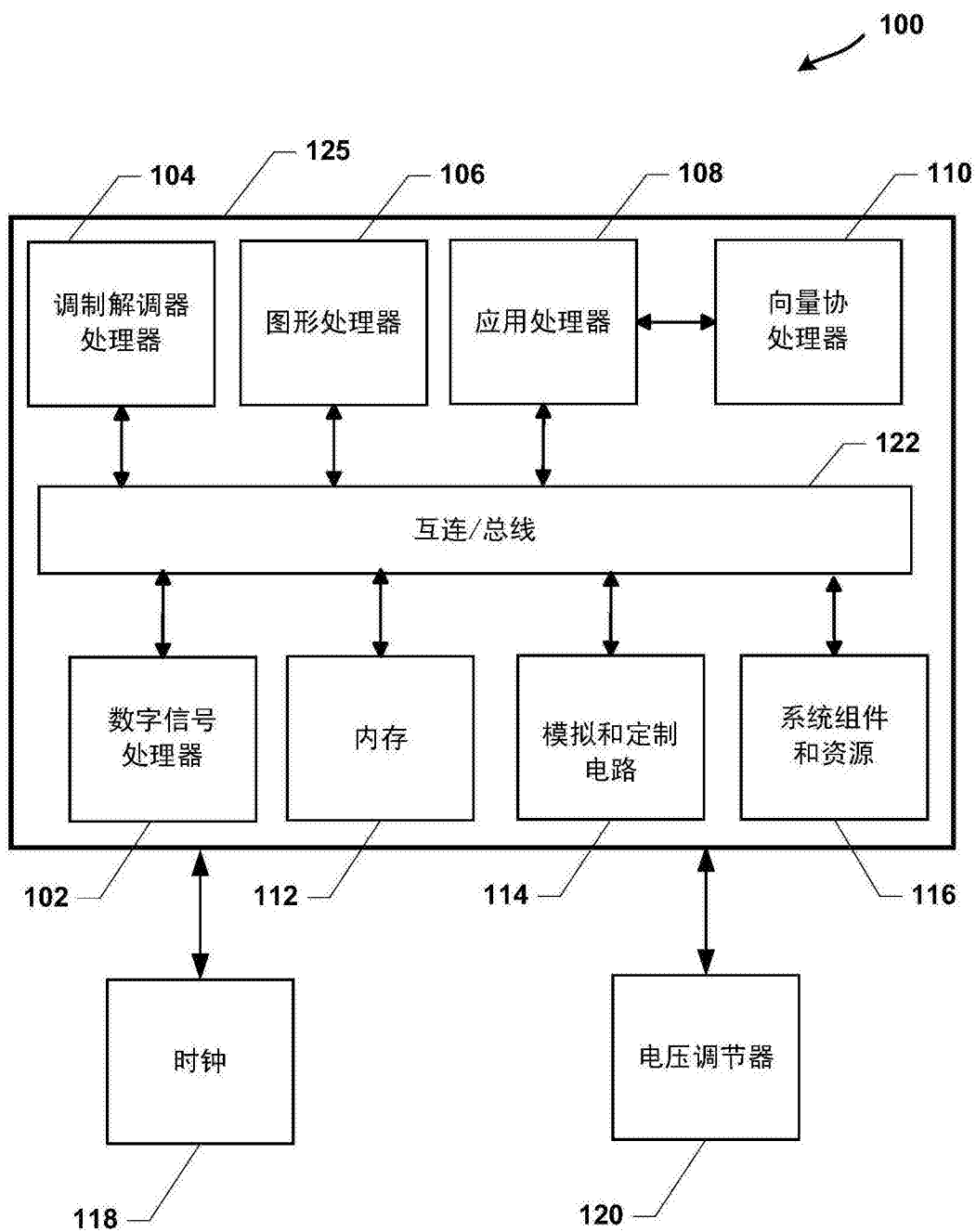


图 1

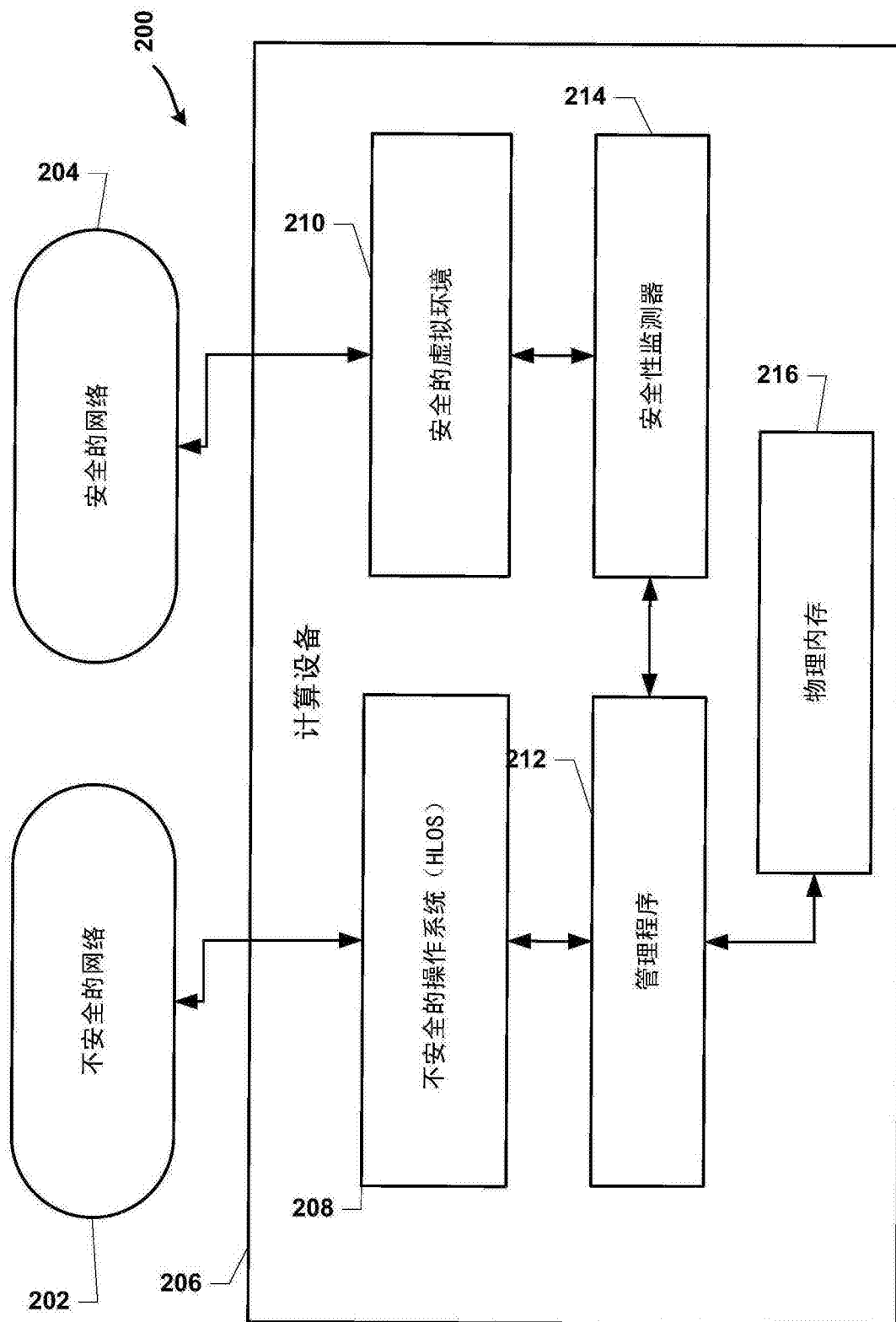


图 2

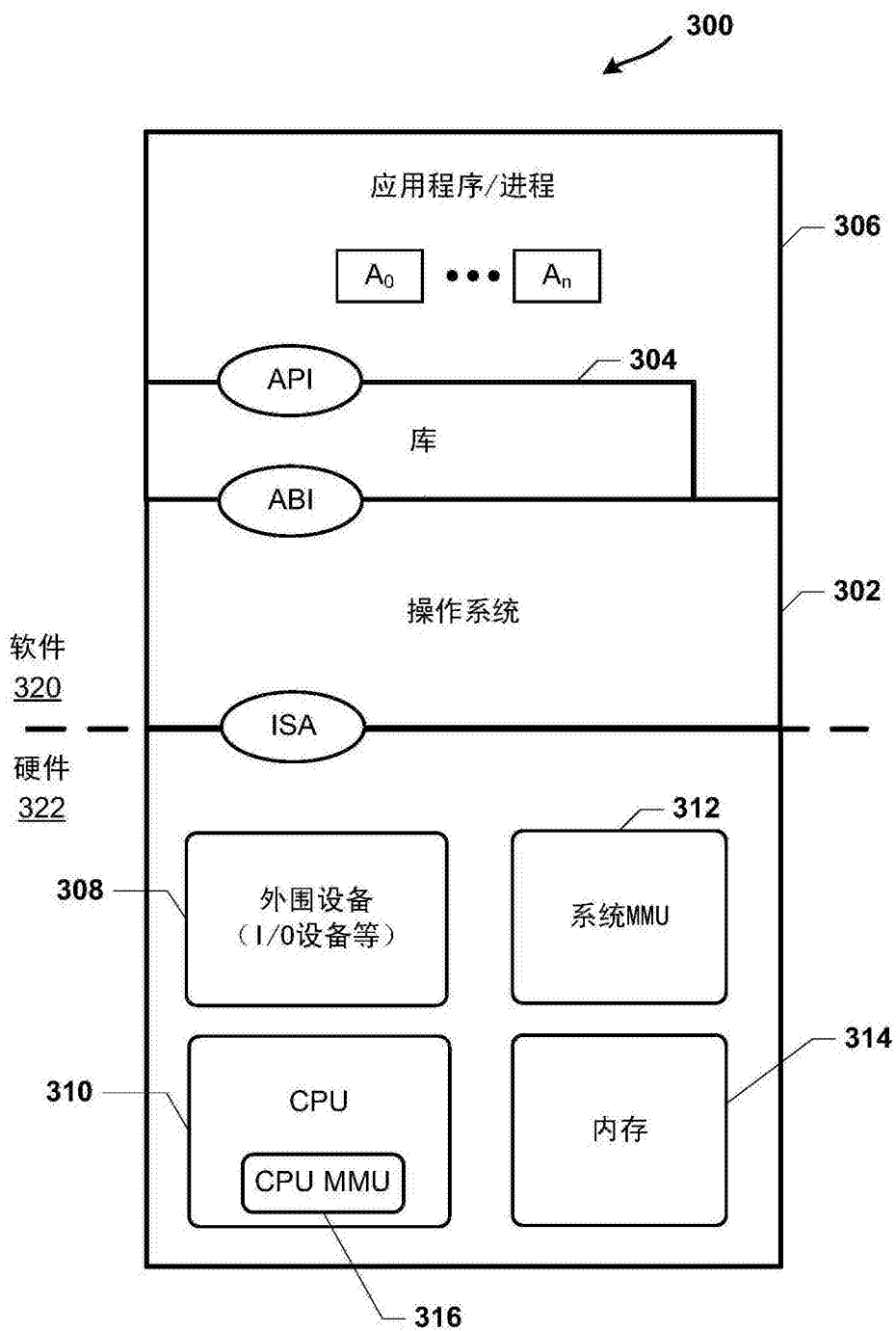


图 3

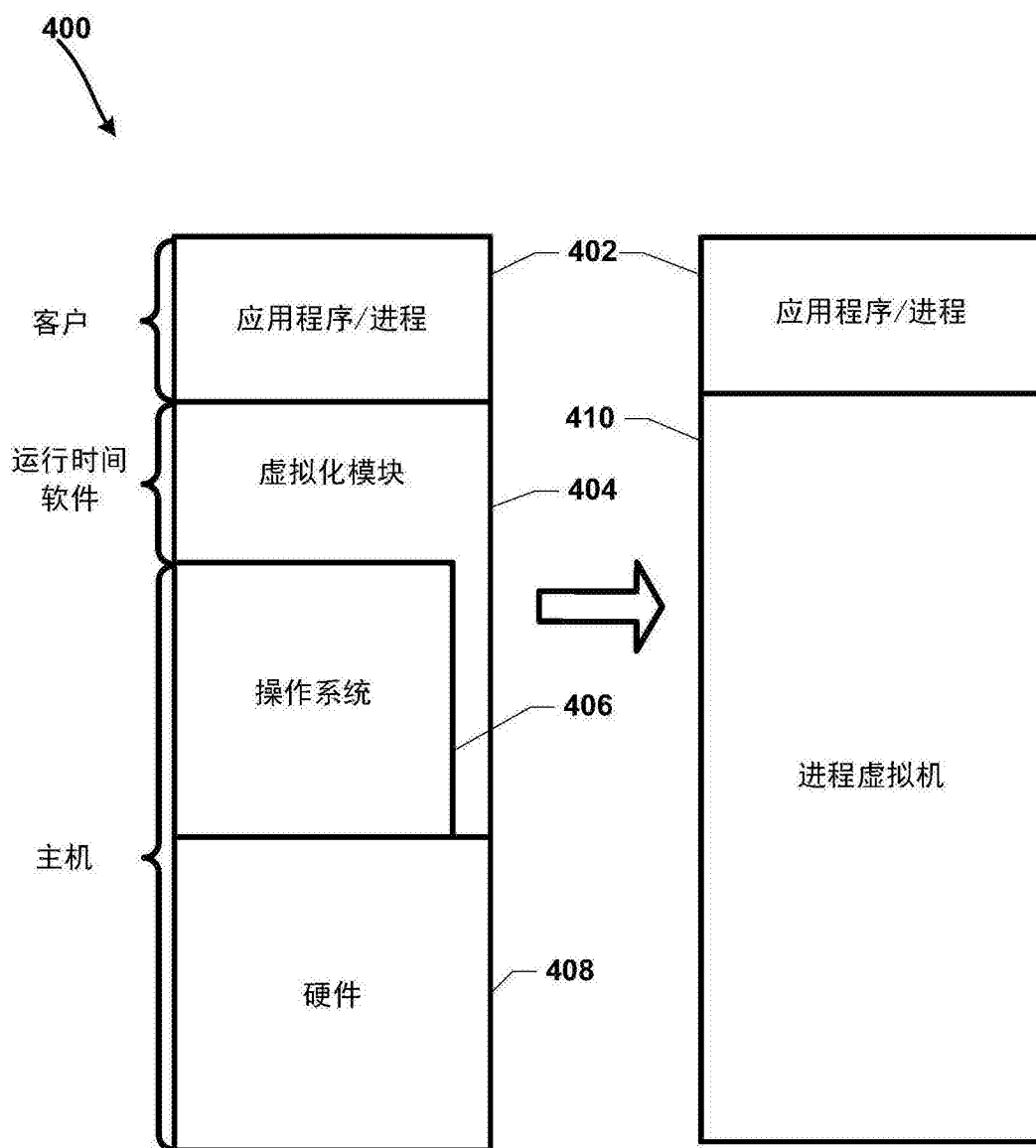


图 4

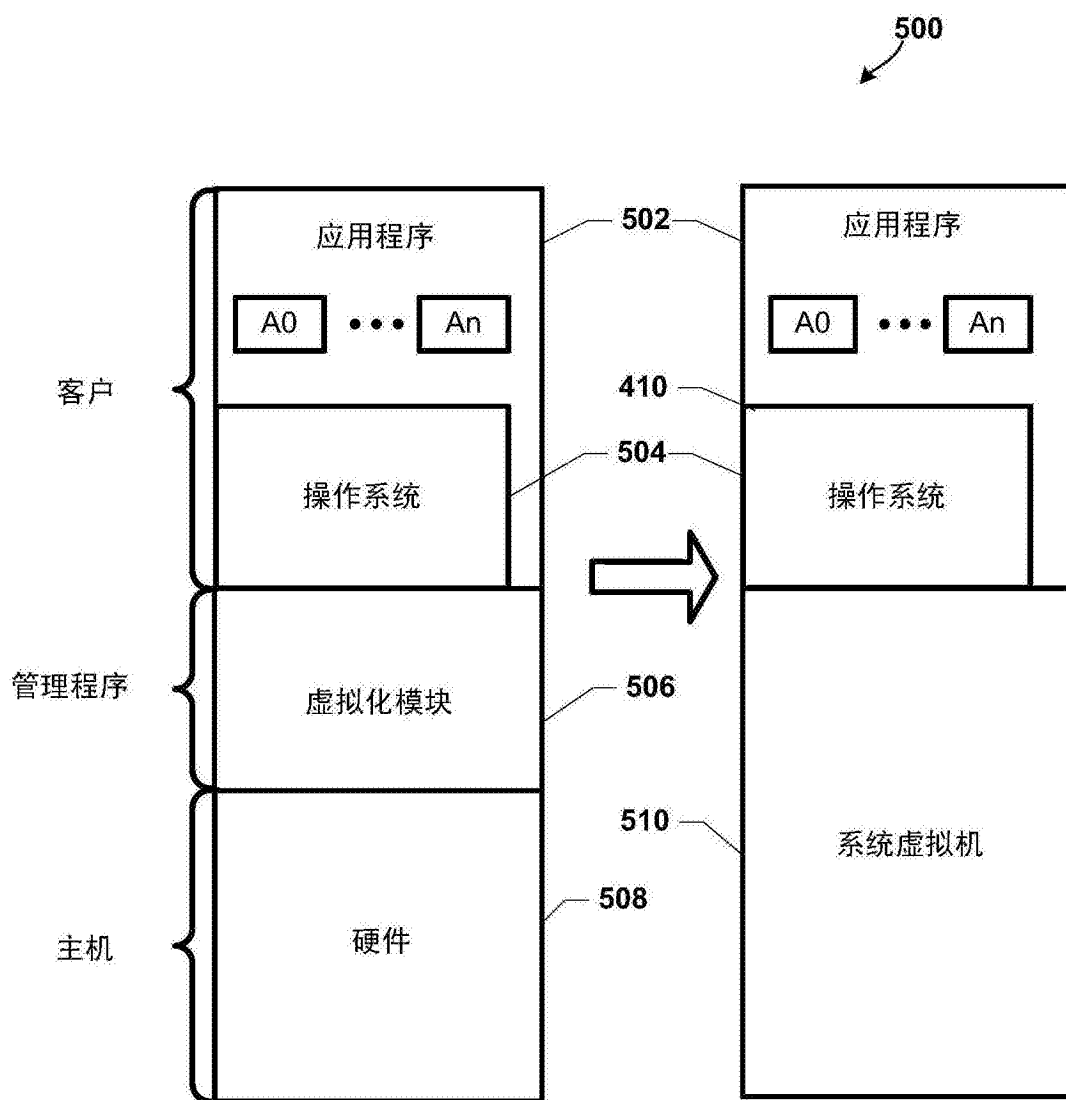


图 5

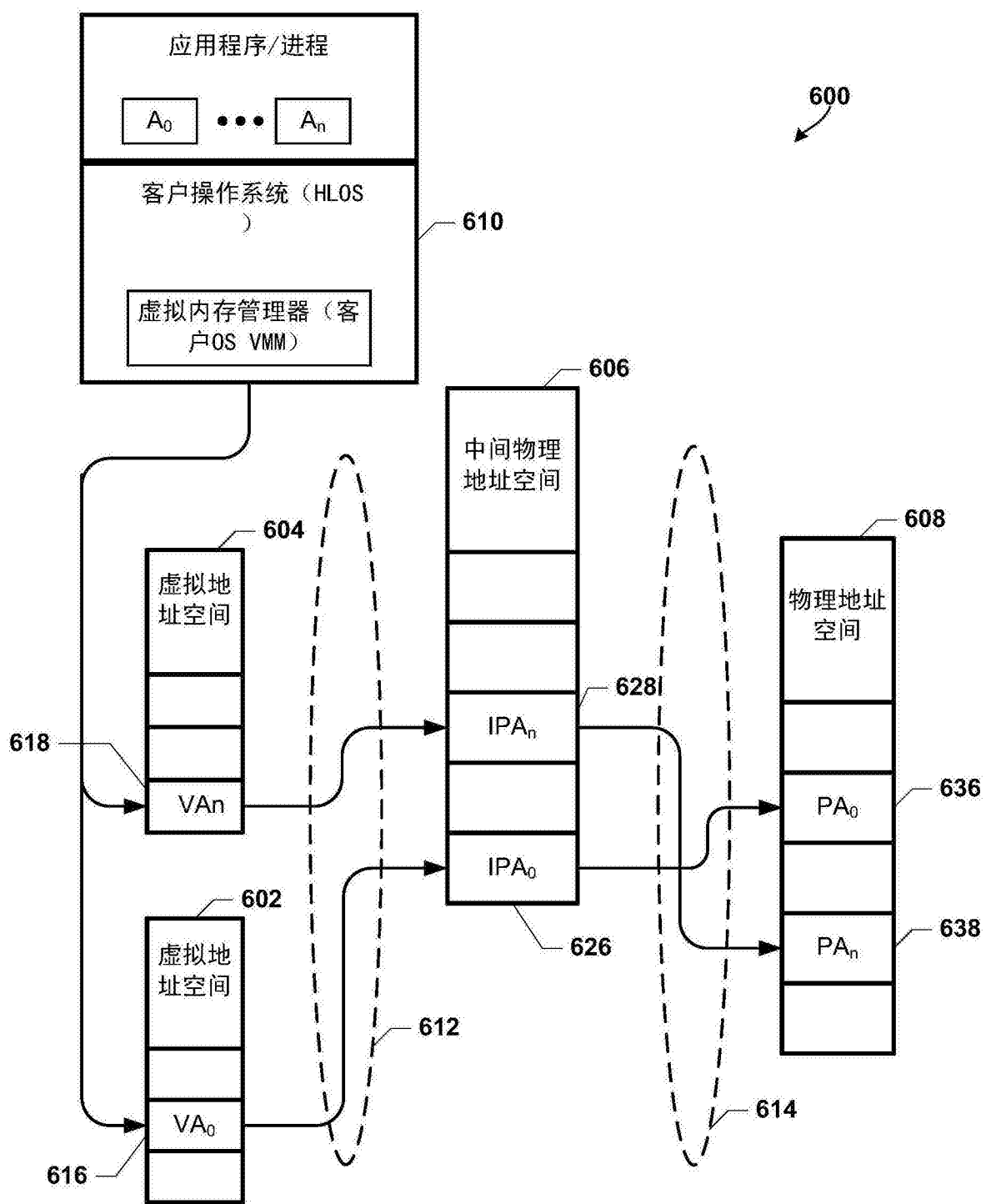


图 6

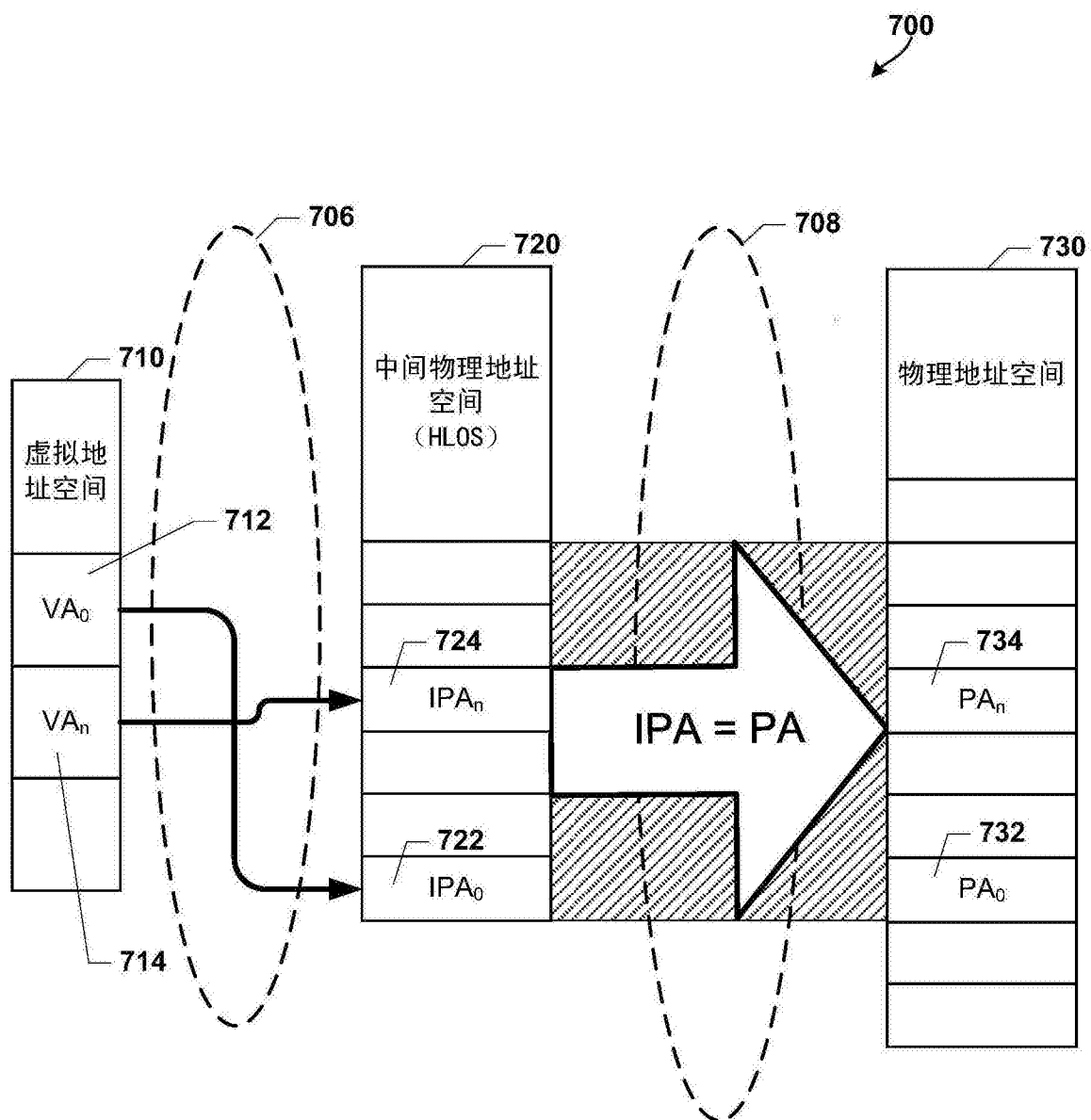


图 7

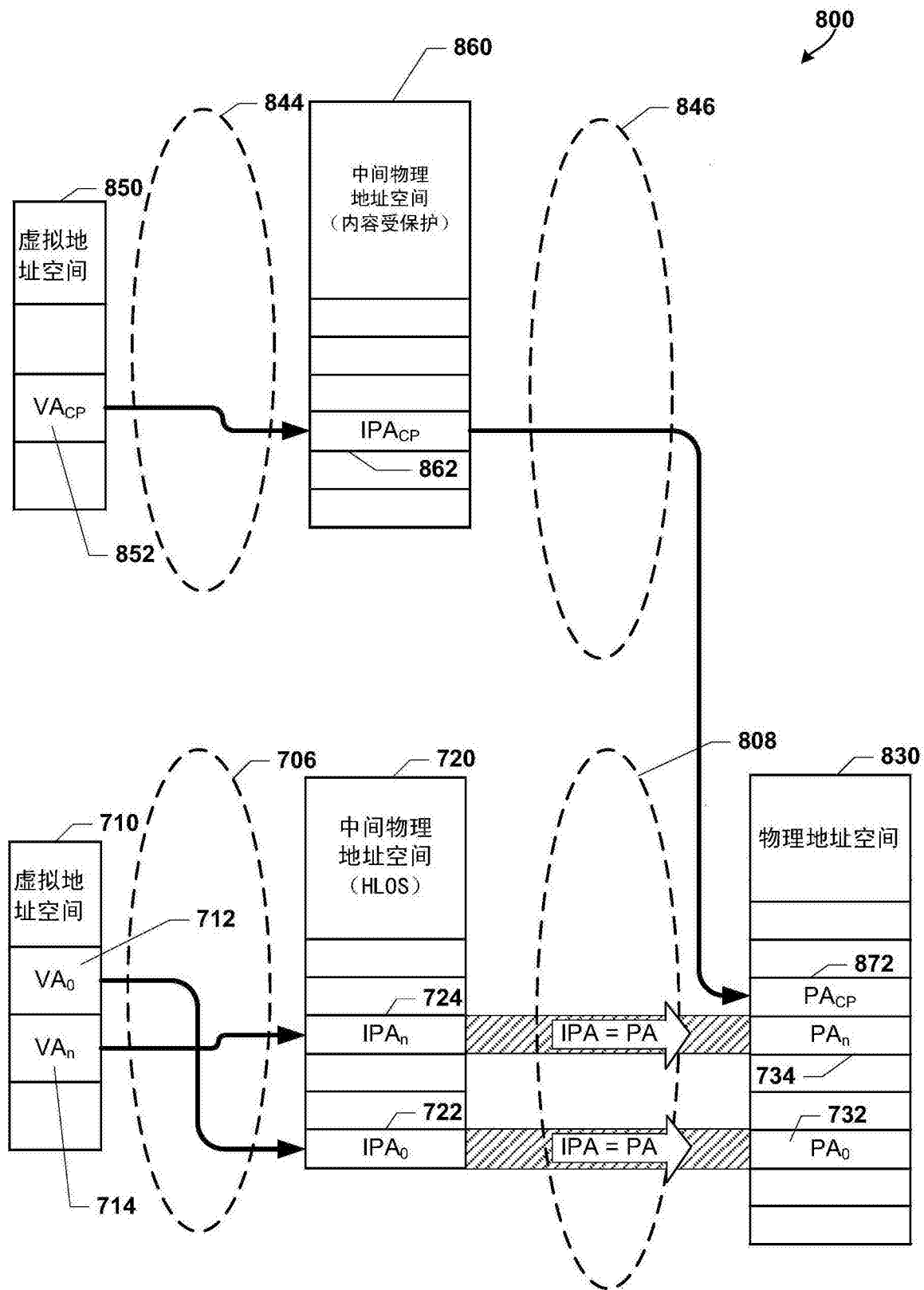


图 8

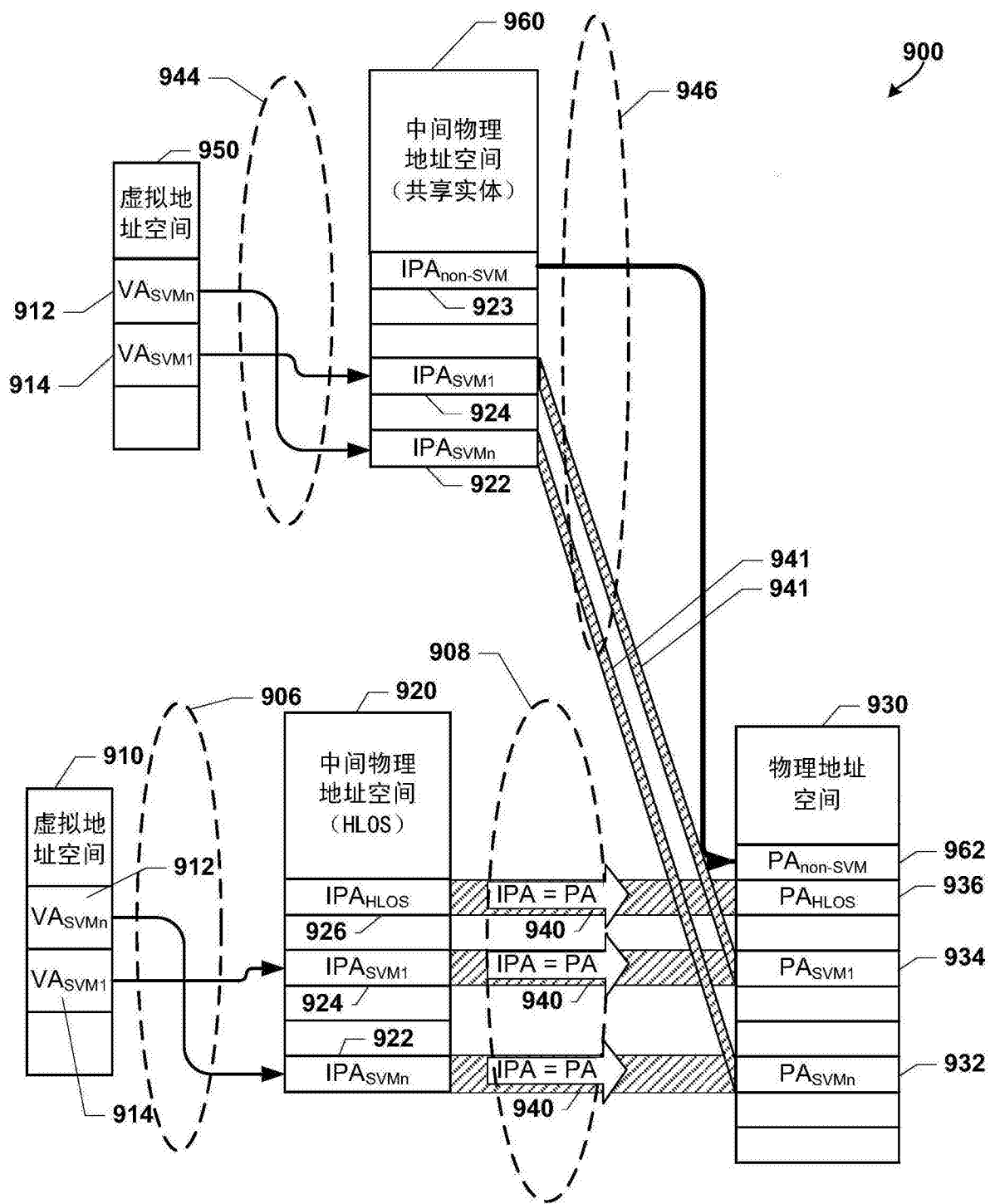


图 9

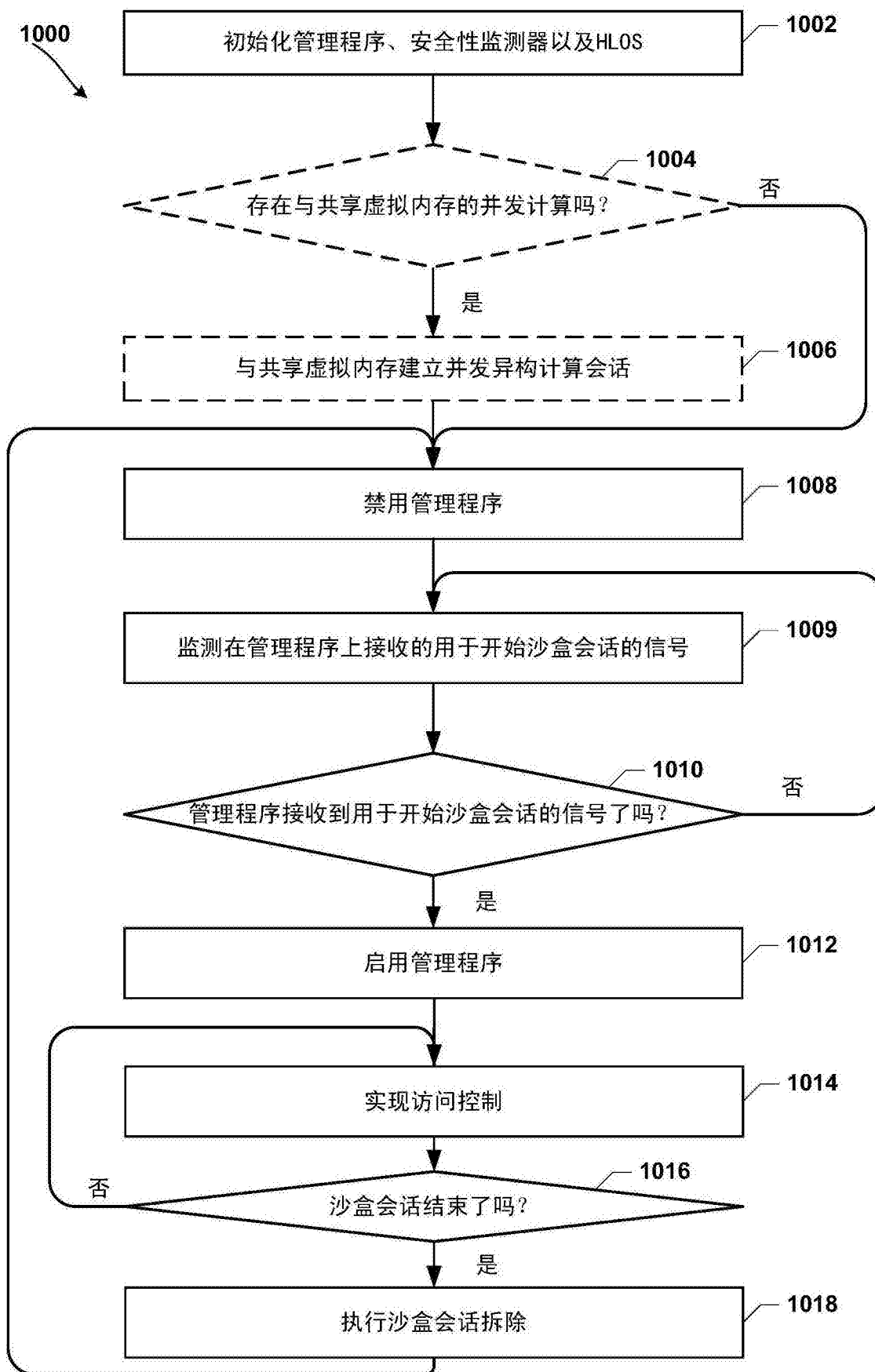


图 10

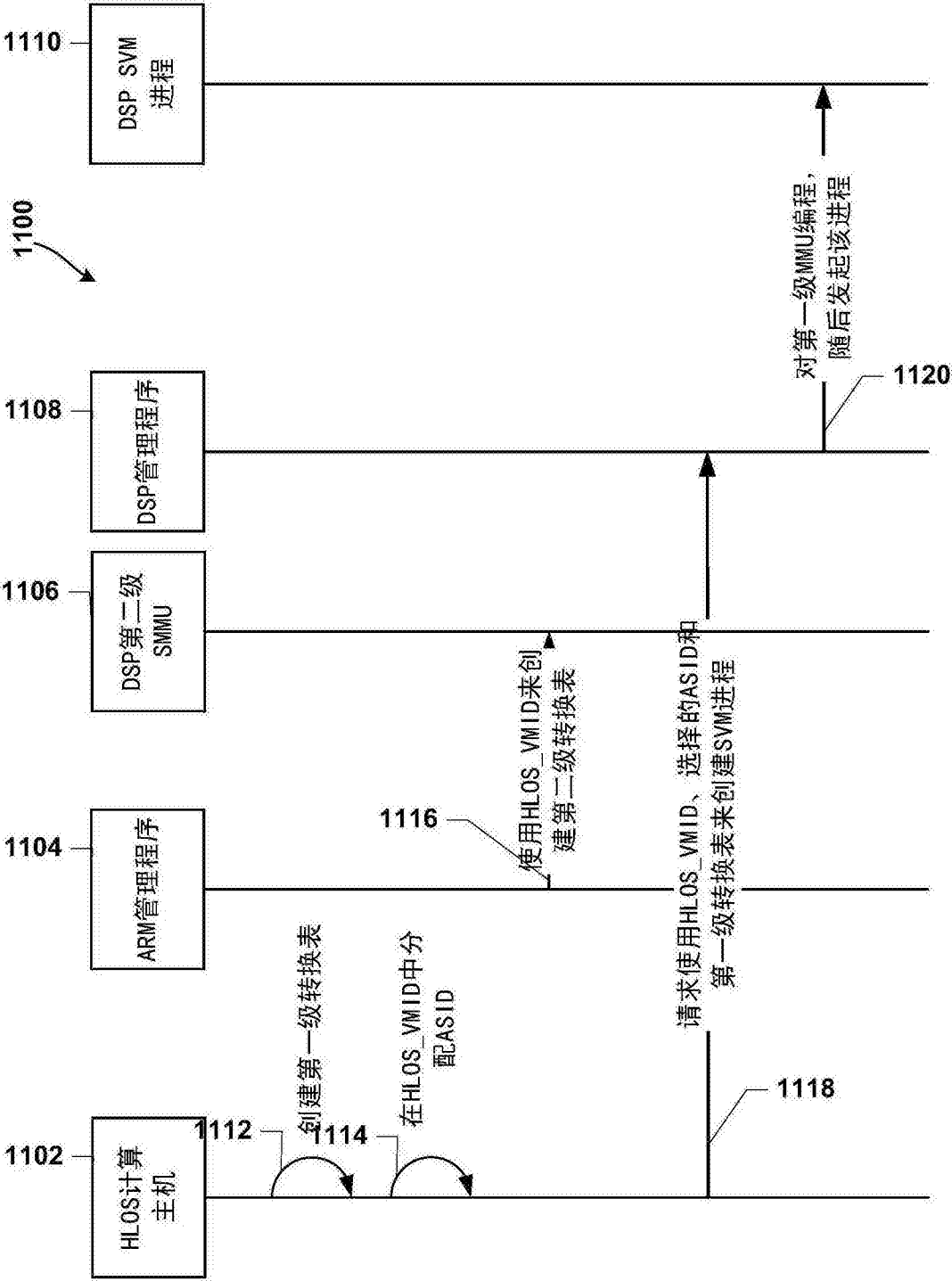


图 11

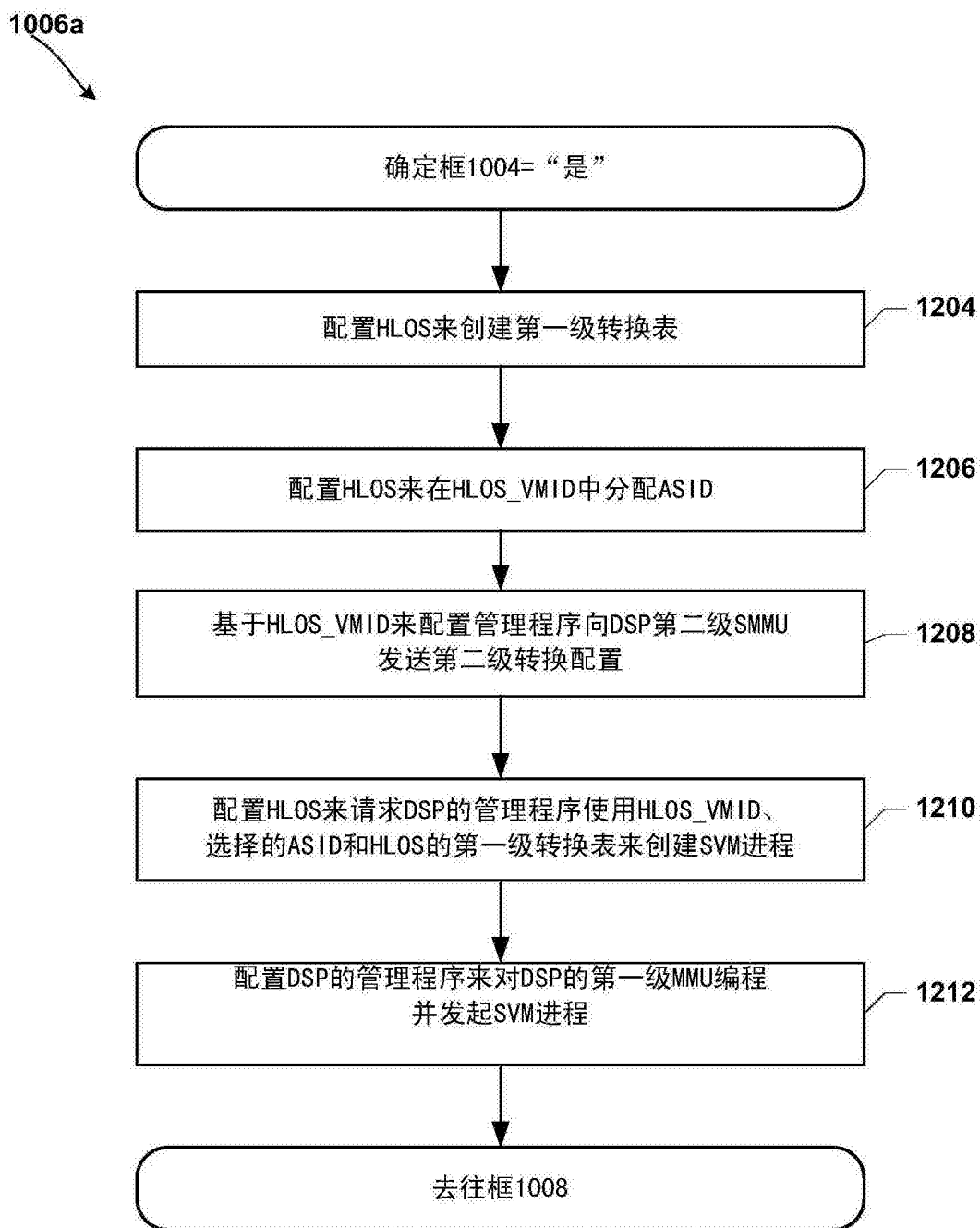


图 12

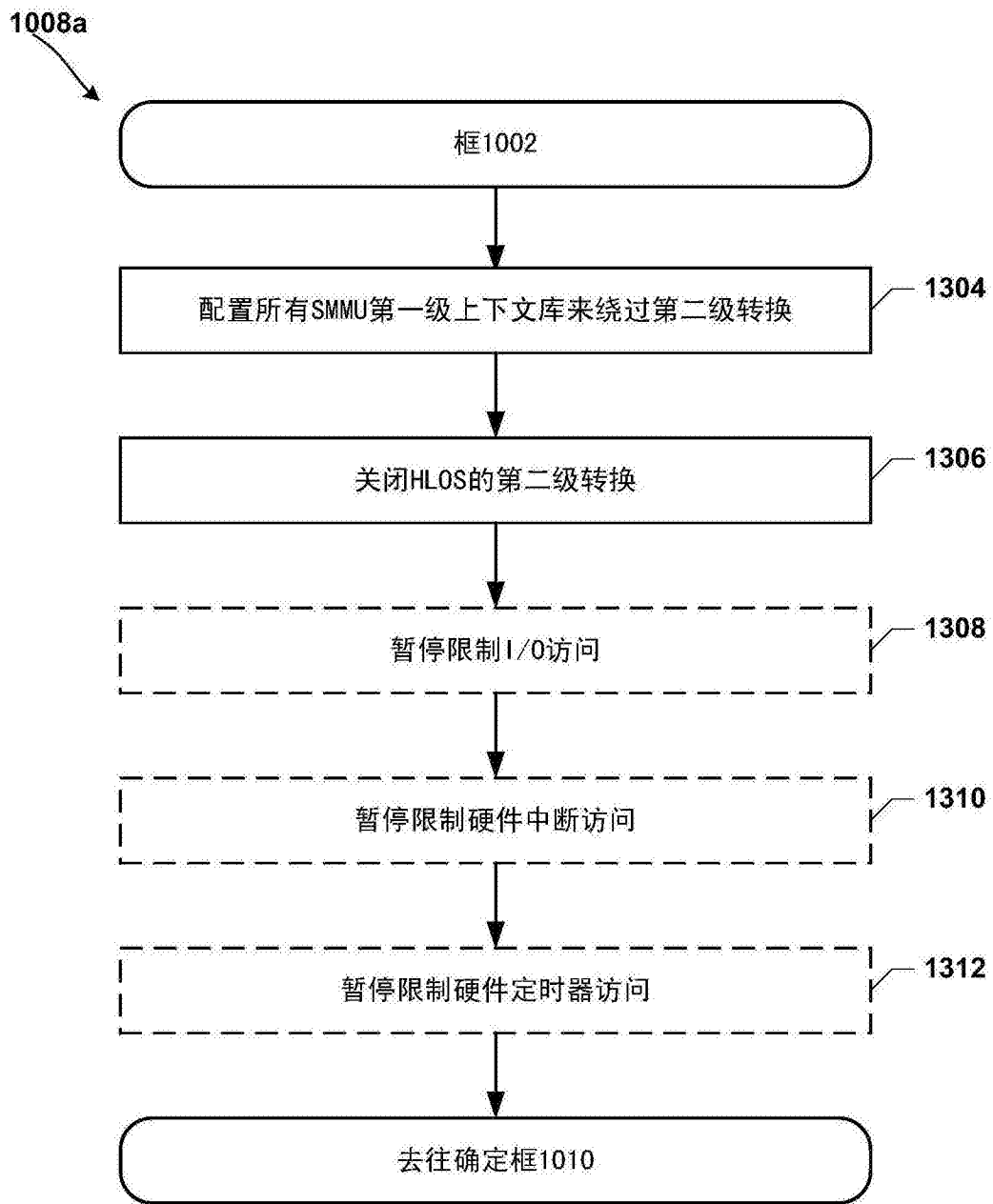


图 13

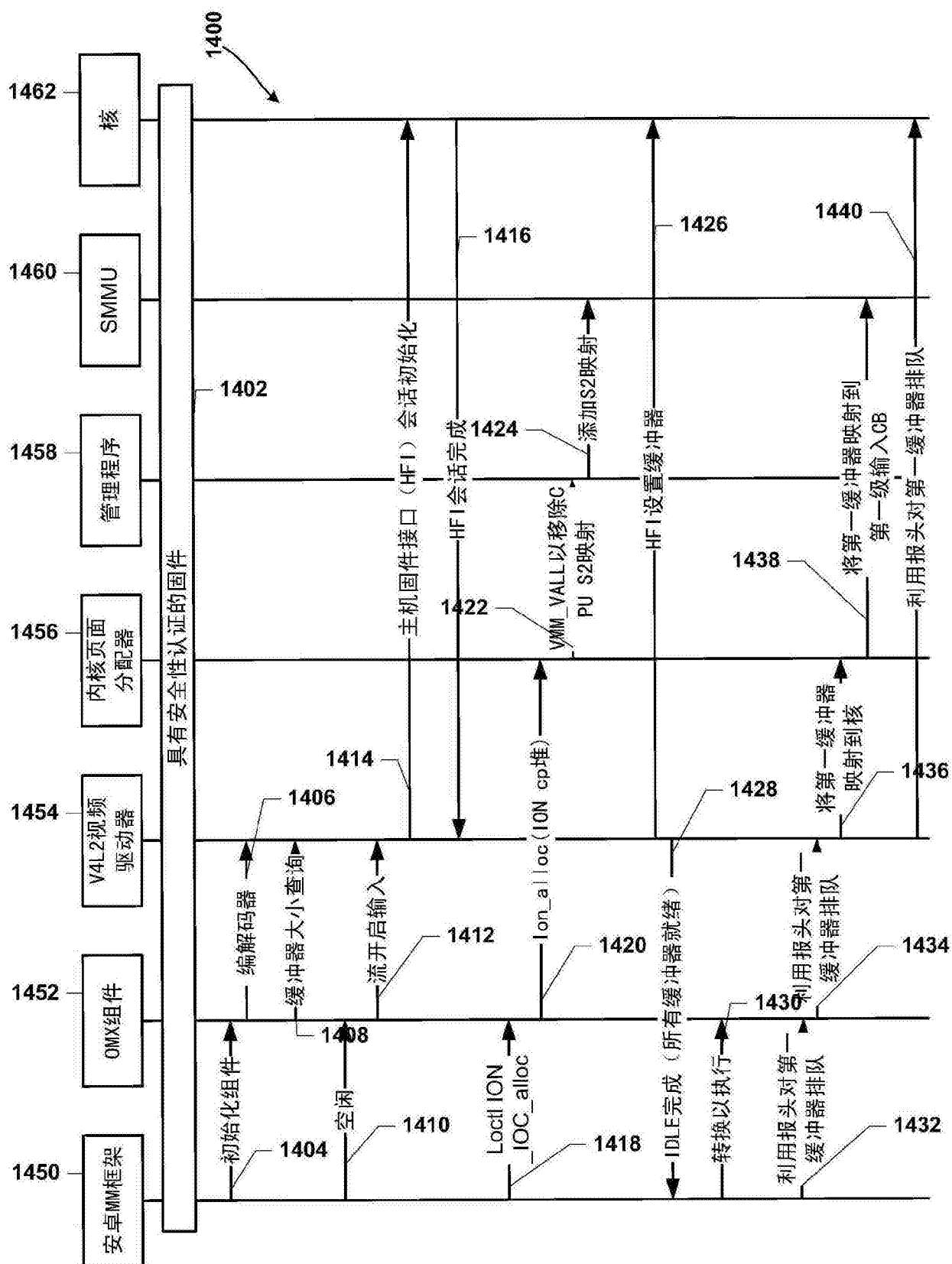


图 14

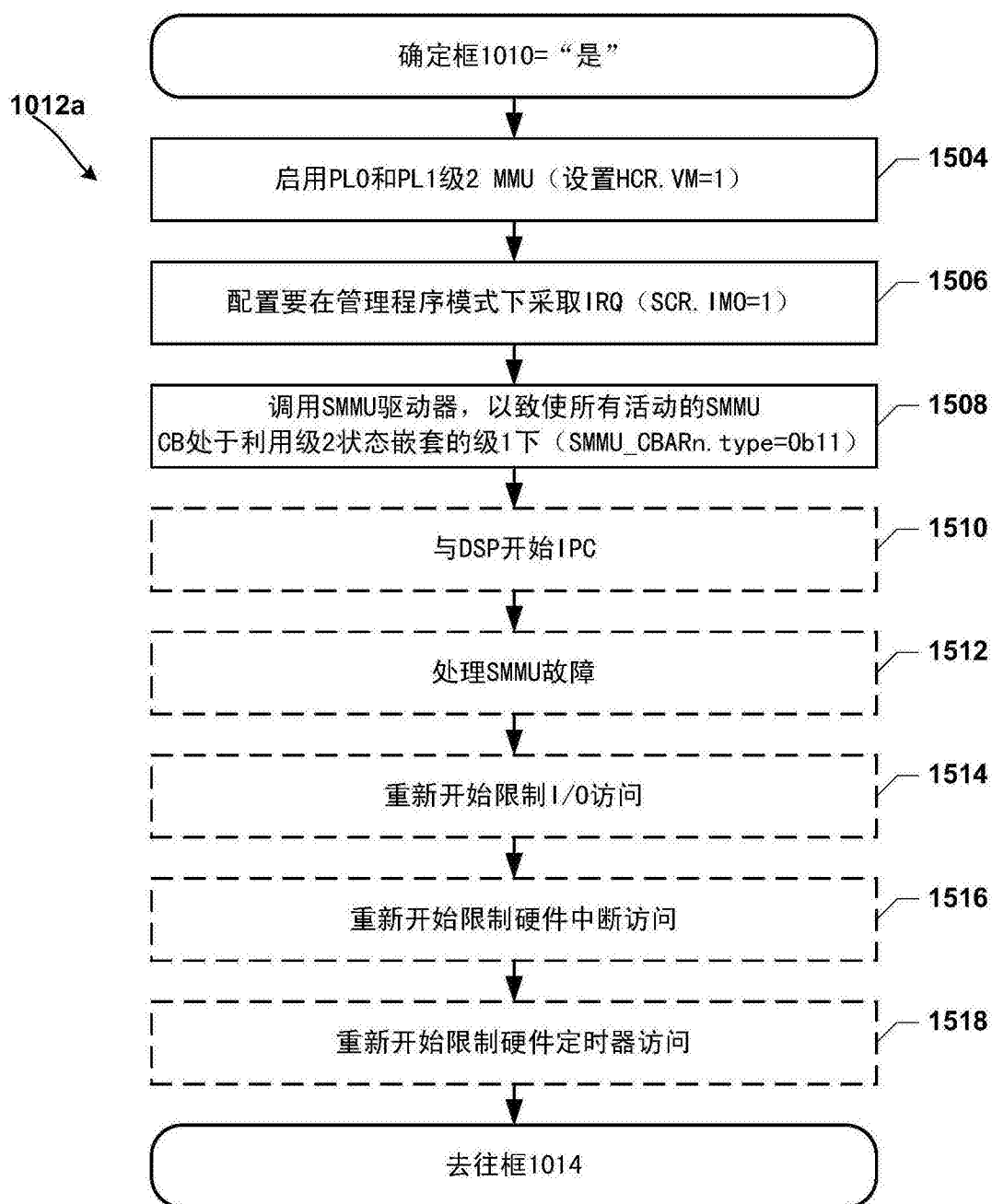


图 15

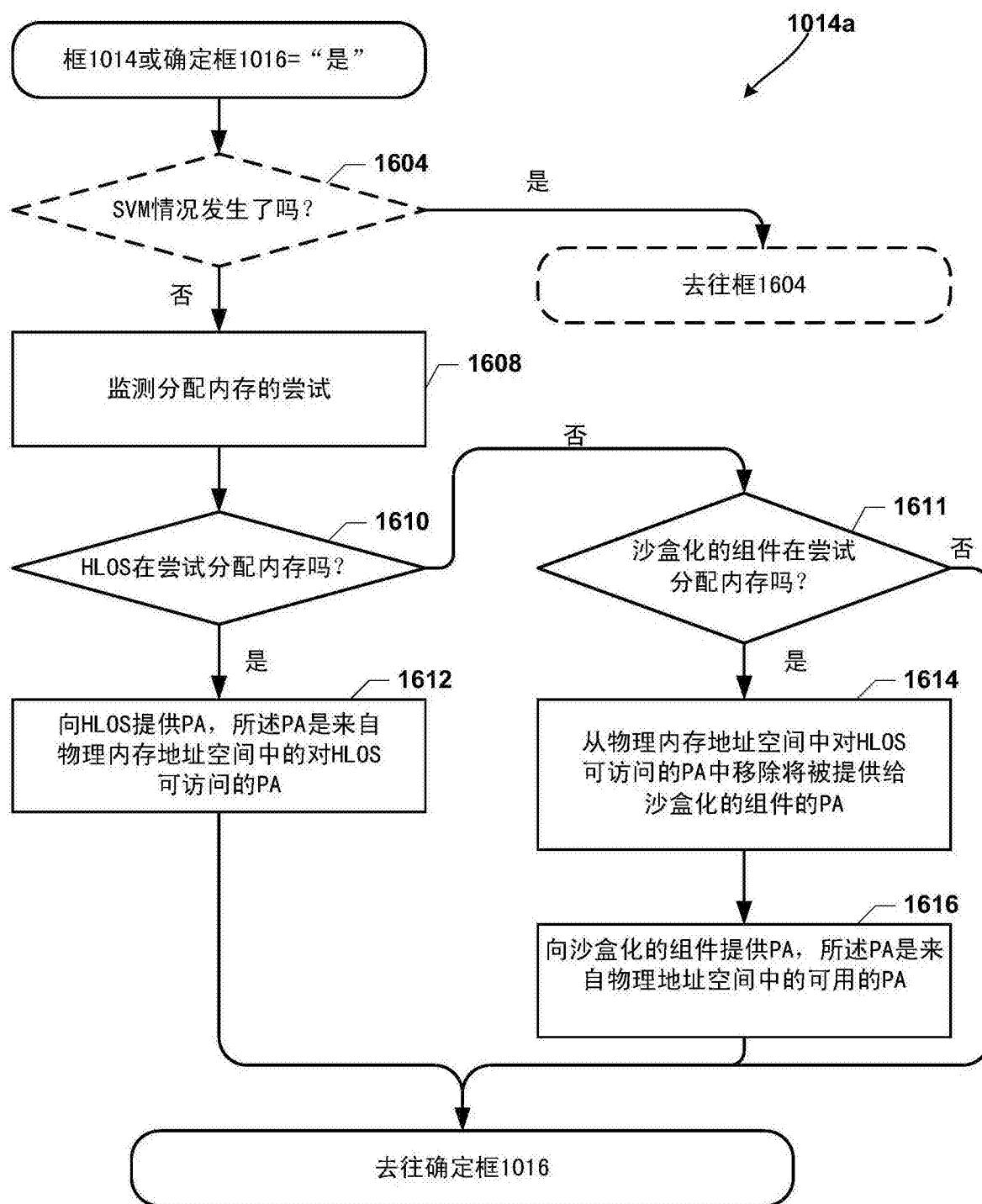


图 16A

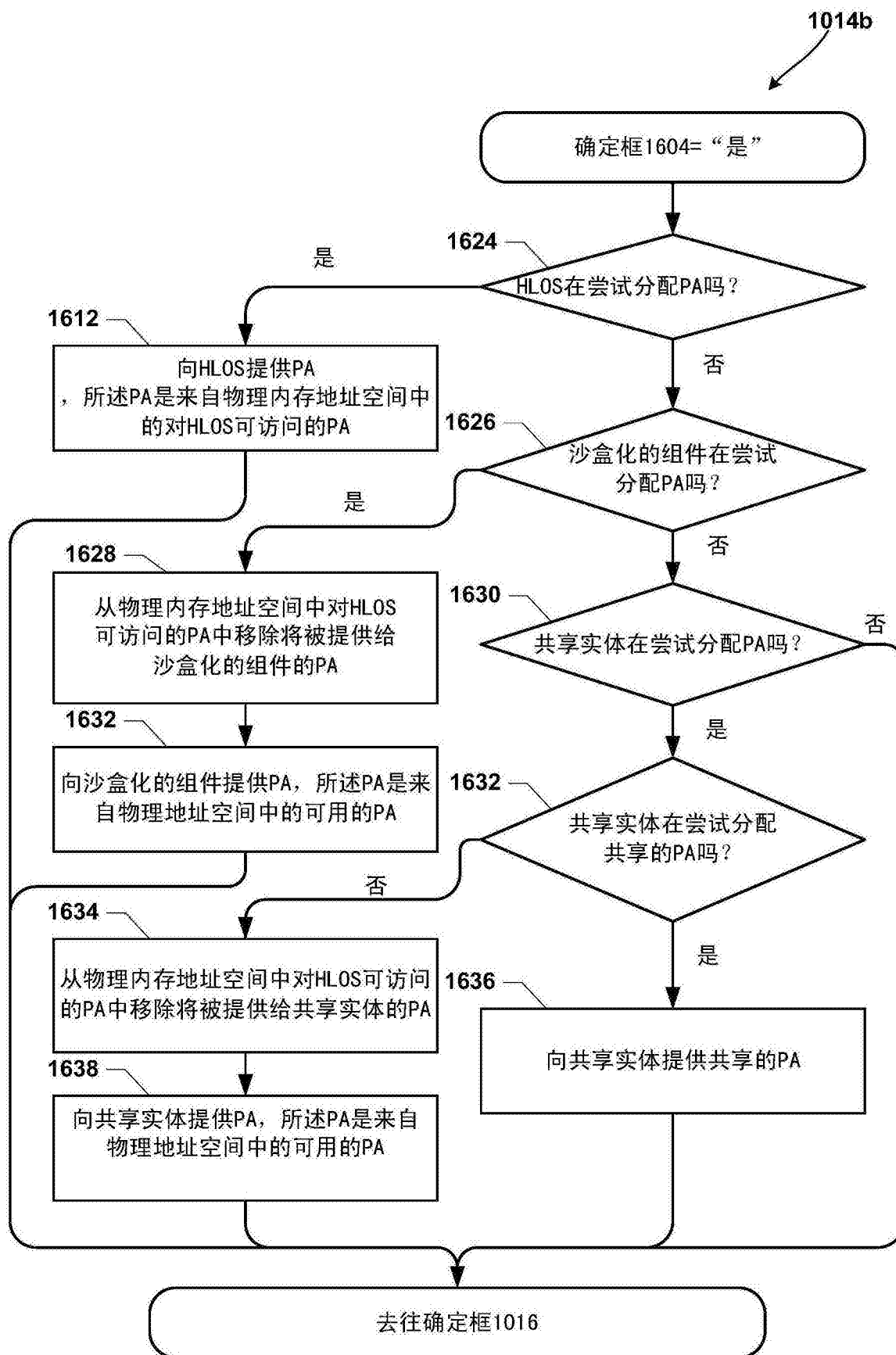


图 16B

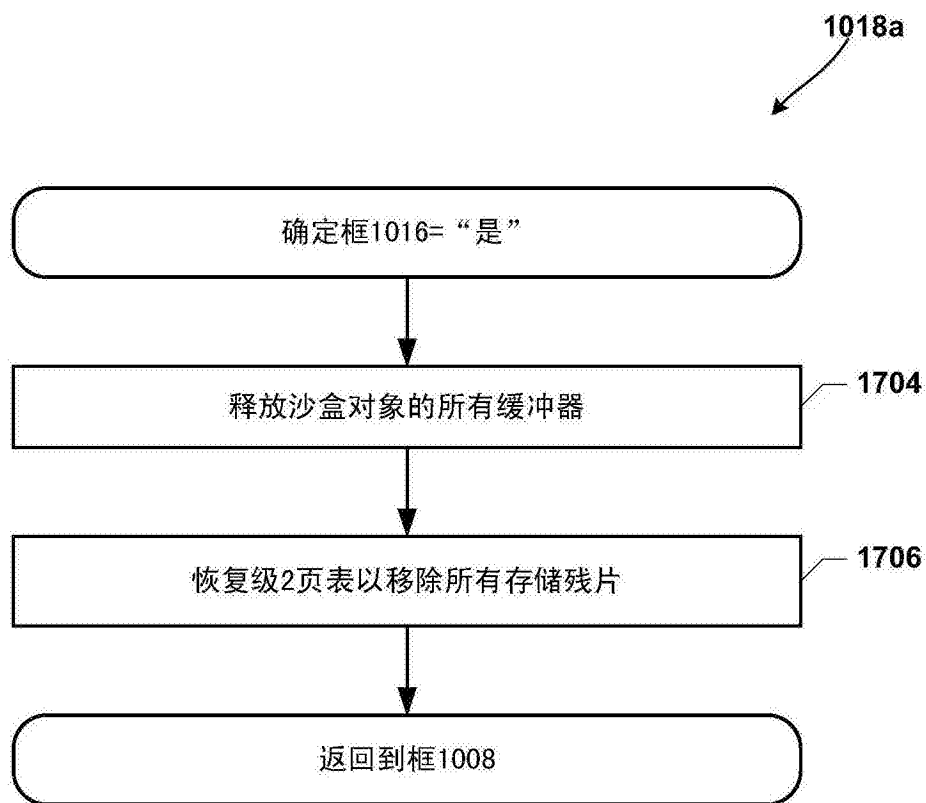


图 17

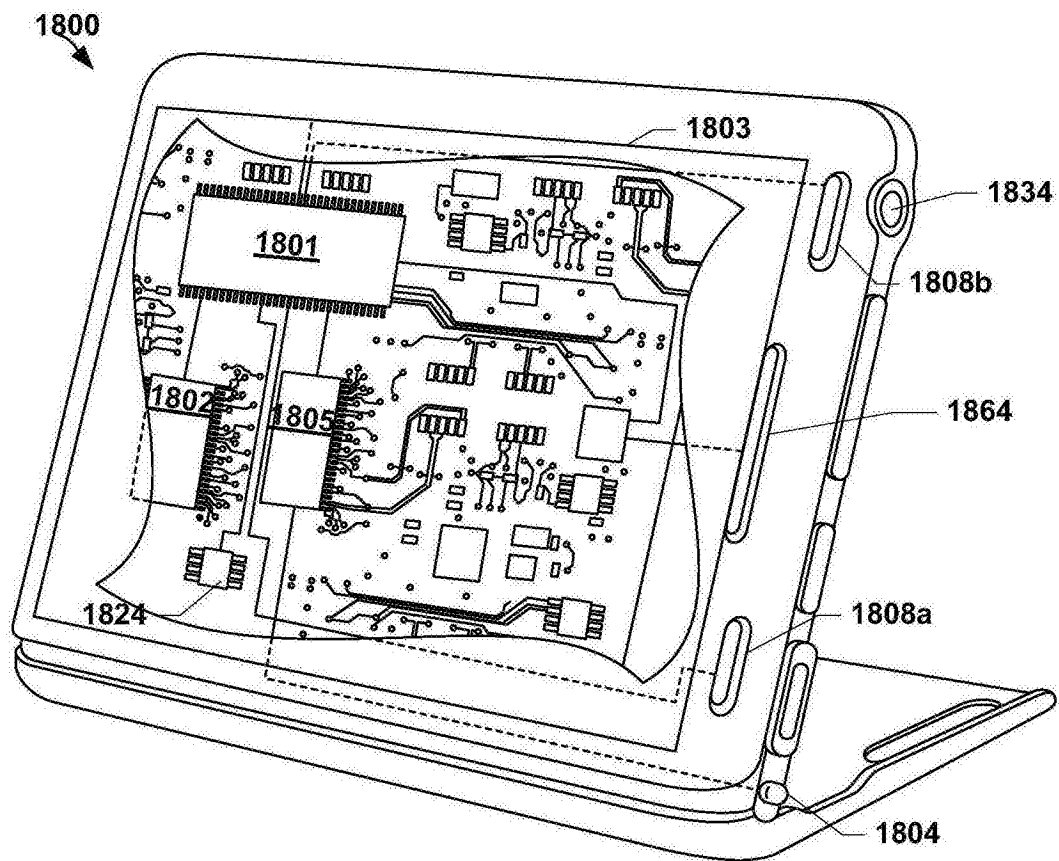


图 18

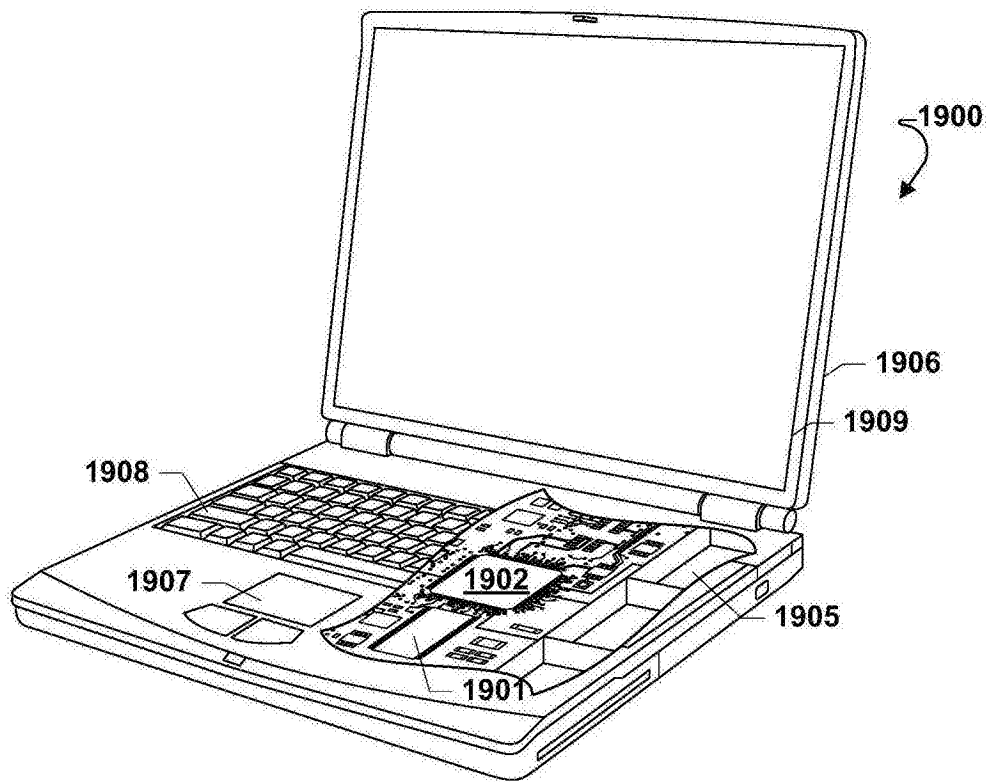


图 19