



(12) 发明专利申请

(10) 申请公布号 CN 103518190 A

(43) 申请公布日 2014. 01. 15

(21) 申请号 201280022612. X

托马斯·萨托里乌斯

(22) 申请日 2012. 05. 21

(74) 专利代理机构 北京律盟知识产权代理有
限责任公司 11287

(30) 优先权数据

61/491, 075 2011. 05. 27 US

13/181, 712 2011. 07. 13 US

代理人 宋献涛

(85) PCT国际申请进入国家阶段日
2013. 11. 11

(51) Int. Cl.

G06F 9/44 (2006. 01)

(86) PCT国际申请的申请数据
PCT/US2012/038797 2012. 05. 21

(87) PCT国际申请的公布数据
W02012/166410 EN 2012. 12. 06

(71) 申请人 高通股份有限公司
地址 美国加利福尼亚州

(72) 发明人 路易斯·塞泽
穆罕默德·H·雷夏迪

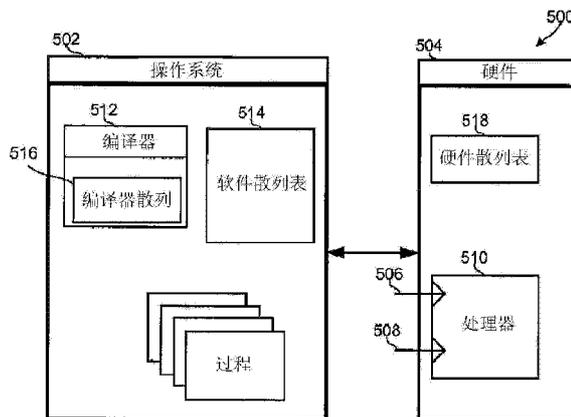
权利要求书7页 说明书11页 附图8页

(54) 发明名称

对于动态语言的散列表的硬件支持

(57) 摘要

各方面使计算装置能够在硬件中执行传统上基于软件的JavaScript®操作。对每一JavaScript®对象进行散列操作进入可存储在软件中的主控散列表中。所述软件散列表的一部分可使用专用于散列表处理的特殊指令集寄存器推送到硬件散列表。每当软件过程请求散列表操作(例如,查找)时,检查所述硬件散列表以确定值是否存在于硬件中。如果所述所请求的值在所述硬件散列表中,那么在单一操作步骤中存取所述所请求的值。如果所述所请求的值不在所述硬件散列表中,那么从所述软件中的所述主控散列表提取所述所请求的值,且使用特殊指令集寄存器将所述主控散列表的含有所述所提取的值的一部分推送到所述硬件。



1. 一种用于执行动态软件语言代码的方法,其包括:
编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥;
将所述经编码散列表整数密钥存储在可由软件存取的存储器中;以及
将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中。
2. 根据权利要求1所述的方法,其中所述第一专用硬件指令是将值推送到所述专用硬件散列表实施方案的 SET 指令。
3. 根据权利要求2所述的方法,其进一步包括:
对于需要散列表查找的每一动态软件语言操作,通过在硬件处理器上执行专用硬件指令来确定所述散列表值是否可由所述硬件处理器检索;以及
在所述密钥存储在所述硬件处理器中的情况下执行基于硬件的散列表查找以传回所述对应的散列表整数密钥。
4. 根据权利要求3所述的方法,其进一步包括:
在所述密钥不在所述硬件处理器上的情况下执行基于软件的散列表查找以检索所述所请求的散列表值和存储在所述软件可存取存储器中的所述经编码散列表整数密钥的一部分;以及
通过将经编码散列表整数密钥的所述所检索部分经由第二专用硬件指令加载到所述硬件处理器的专用存储器中来更新所述专用硬件散列表实施方案。
5. 根据权利要求4所述的方法,其中所述第二专用硬件指令为从所述硬件处理器的所述专用存储器检索值的 GET 指令。
6. 根据权利要求5所述的方法,其中将散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一动态软件语言操作请求执行所述经编码散列表整数密钥的一部分的加载。
7. 根据权利要求5所述的方法,其中:
所述动态软件语言代码为JavaScript®代码;且
编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥包括编译JavaScript 代码以将 JavaScript 散列表字符串密钥编码为散列表整数密钥。
8. 根据权利要求7所述的方法,其中将JavaScript®散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一JavaScript®操作请求执行所述经编码散列表整数密钥的一部分的加载。
9. 根据权利要求1所述的方法,其中所述第一专用硬件指令为通过将值存储在存储器映射位置中而将所述值推送到所述专用硬件散列表实施方案的 STORE 指令。
10. 根据权利要求1所述的方法,其中将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中包括在耦合到主处理器的协处理器上将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中。
11. 根据权利要求1所述的方法,其中编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥是由计算装置的第一处理器执行,且将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中是由所述计算装

置的第二处理器执行。

12. 根据权利要求 11 所述的方法,其中所述计算装置的所述第二处理器为专门散列表协处理器。

13. 根据权利要求 12 所述的方法,其进一步包括:

对于需要散列表查找的每一动态软件语言操作,通过在所述专门散列表协处理器上执行第二专用硬件指令而确定是否可检索所述散列表值;以及

在确定可检索所述散列表值的情况下在所述专门散列表协处理器上执行基于硬件的散列表查找以接收所述计算装置的所述第一处理器上的所述对应的散列表整数密钥。

14. 一种计算装置,其包括:

存储器;以及

耦合到所述存储器的处理器,所述处理器配置有处理器可执行指令以执行包括以下各项的操作:

编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥;

将所述经编码散列表整数密钥存储在所述存储器中使得所述经编码散列表整数密钥可由软件存取;以及

将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中。

15. 根据权利要求 14 所述的计算装置,其中所述存储器为所述处理器内的散列表存储器。

16. 根据权利要求 14 所述的计算装置,其中所述存储器在所述处理器外部。

17. 根据权利要求 14 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得所述第一专用硬件指令是将值推送到所述专用硬件散列表实施方案的 SET 指令。

18. 根据权利要求 17 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行进一步包括以下各项的操作:

对于需要散列表查找的每一动态软件语言操作,通过在硬件处理器上执行专用硬件指令来确定所述散列表值是否可由所述硬件处理器检索;以及

在所述密钥存储在所述硬件处理器中的情况下执行基于硬件的散列表查找以传回所述对应的散列表整数密钥。

19. 根据权利要求 18 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行进一步包括以下各项的操作:

在所述密钥不在所述硬件处理器上的情况下执行基于软件的散列表查找以检索所述所请求的散列表值和存储在所述软件可存取存储器中的所述经编码散列表整数密钥的一部分;以及

通过将经编码散列表整数密钥的所述所检索部分经由第二专用硬件指令加载到所述硬件处理器的专用存储器中来更新所述专用硬件散列表实施方案。

20. 根据权利要求 19 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得所述第二专用硬件指令为从所述硬件处理器的所述专用存储器检索值的 GET 指令。

21. 根据权利要求 20 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得所述将散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一动态软件语言操作请求执行所述经编码散列表整数密钥的一部分的加载。

22. 根据权利要求 20 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得:

所述动态软件语言代码为 JavaScript® 代码;且

编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥包括编译 JavaScript 代码以将 JavaScript 散列表字符串密钥编码为散列表整数密钥。

23. 根据权利要求 22 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得将 JavaScript® 散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一 JavaScript® 操作请求执行所述经编码散列表整数密钥的一部分的加载。

24. 根据权利要求 14 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得所述第一专用硬件指令为通过将值存储在存储器映射位置中而将所述值推送到所述专用硬件散列表实施方案的 STORE 指令。

25. 根据权利要求 14 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行操作使得将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中包括在耦合到主处理器的协处理器上将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中。

26. 根据权利要求 14 所述的计算装置,其中所述处理器包括第一处理器和第二处理器,且其中所述处理器配置有处理器可执行指令以执行操作使得编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥是由所述第一处理器执行,且将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中是由所述计算装置的所述第二处理器执行。

27. 根据权利要求 26 所述的计算装置,其中所述第二处理器为专门散列表协处理器。

28. 根据权利要求 27 所述的计算装置,其中所述处理器配置有处理器可执行指令以执行进一步包括以下各项的操作:

对于需要散列表查找的每一动态软件语言操作,通过在所述专门散列表协处理器上执行第二专用硬件指令而确定是否可检索所述散列表值;以及

在确定可检索所述散列表值的情况下在所述专门散列表协处理器上执行基于硬件的散列表查找以接收所述计算装置的所述第一处理器上的所述对应的散列表整数密钥。

29. 一种计算装置,其包括:

用于编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥的装置;

用于将所述经编码散列表整数密钥存储在可由软件存取的存储器中的装置;以及

用于将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置。

30. 根据权利要求 29 所述的计算装置,其中所述存储器为所述计算装置的处理器内的散列表存储器。

31. 根据权利要求 29 所述的计算装置,其中所述存储器在所述计算装置的处理器的外部。

32. 根据权利要求 29 所述的计算装置,其中用于将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置包括用于将所述经编码散列表整数密钥的一部分经由将值推送到所述专用硬件散列表实施方案的 SET 指令加载到专用硬件散列表实施方案中的装置。

33. 根据权利要求 32 所述的计算装置,其进一步包括:

用于对于需要散列表查找的每一动态软件语言操作通过在硬件处理器上执行专用硬件指令来确定所述散列表值是否可由所述硬件处理器检索的装置;以及

用于在所述密钥存储在所述硬件处理器中的情况下执行基于硬件的散列表查找以传回所述对应的散列表整数密钥的装置。

34. 根据权利要求 33 所述的计算装置,其进一步包括:

用于在所述密钥不在所述硬件处理器上的情况下执行基于软件的散列表查找以检索所述所请求的散列表值和存储在所述软件可存取存储器中的所述经编码散列表整数密钥的一部分的装置;以及

用于通过将经编码散列表整数密钥的所述所检索部分经由第二专用硬件指令加载到所述硬件处理器的专用存储器中来更新所述专用硬件散列表实施方案的装置。

35. 根据权利要求 34 所述的计算装置,其中用于通过将经编码散列表整数密钥的所述所检索部分经由第二专用硬件指令加载到所述硬件处理器的专用存储器中来更新所述专用硬件散列表实施方案的装置包括用于通过将经编码散列表整数密钥的所述所检索部分经由从所述硬件处理器的专用存储器检索值的 GET 指令加载到所述硬件处理器的所述专用存储器中来更新所述专用硬件散列表实施方案的装置。

36. 根据权利要求 35 所述的计算装置,其中:

用于编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥的装置包括用于在编译时间正好编译所述动态软件语言代码一次以将散列表字符串密钥编码为散列表整数密钥的装置;以及

用于将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置包括用于依据每一动态软件语言操作请求将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置。

37. 根据权利要求 35 所述的计算装置,其中:

所述动态软件语言代码为 JavaScript® 代码;且

用于编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥的装置包括用于编译 JavaScript 代码以将 JavaScript 散列表字符串密钥编码为散列表整数密钥的装置。

38. 根据权利要求 37 所述的计算装置,其中:

用于将 JavaScript® 散列表字符串密钥编码为散列表整数密钥的装置包括用于在编译时间正好将 JavaScript® 散列表字符串密钥编码为散列表整数密钥一次的装置;

且

用于将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置包括用于依据每一 JavaScript® 操作请求将所述经编码散列表整

数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置。

39. 根据权利要求 29 所述的计算装置,其中所述第一专用硬件指令为通过将值存储在存储器映射位置中而将所述值推送到所述专用硬件散列表实施方案的 STORE 指令。

40. 根据权利要求 29 所述的计算装置,其中用于将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中的装置包括用于在耦合到主处理器的协处理器上将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中的装置。

41. 根据权利要求 29 所述的计算装置,其中:

第一处理器;

第二处理器;

用于编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥的装置包括用于通过所述第一处理器编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥的装置;以及

用于将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置包括用于通过所述第二处理器将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中的装置。

42. 根据权利要求 41 所述的计算装置,其中所述第二处理器为专门散列表协处理器。

43. 根据权利要求 42 所述的计算装置,其进一步包括:

用于对于需要散列表查找的每一动态软件语言操作通过在所述专门散列表协处理器上执行第二专用硬件指令而确定是否可检索所述散列表值的装置;以及

用于在确定可检索所述散列表值的情况下在所述专门散列表协处理器上执行基于硬件的散列表查找以接收所述计算装置的所述第一处理器上的所述对应的散列表整数密钥的装置。

44. 一种非瞬时处理器可读媒体,所述非瞬时处理器可读媒体上存储有处理器可执行指令,所述处理器可执行指令经配置以致使处理器执行包括以下各项的操作:

编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥;

将所述经编码散列表整数密钥存储在可由软件存取的存储器中;以及

将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中。

45. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以在处理器上起作用,其中所述存储器为所述处理器内的散列表存储器。

46. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以在处理器上起作用,其中所述存储器在所述处理器外部。

47. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得所述第一专用硬件指令是将值推送到所述专用硬件散列表实施方案的 SET 指令。

48. 根据权利要求 47 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行进一步包括以下各项的操作:

对于需要散列表查找的每一动态软件语言操作,通过在硬件处理器上执行专用硬件指令来确定所述散列表值是否可由所述硬件处理器检索;以及

在所述密钥存储在所述硬件处理器中的情况下执行基于硬件的散列表查找以传回所述对应的散列表整数密钥。

49. 根据权利要求 48 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行进一步包括以下各项的操作:

在所述密钥不在所述硬件处理器上的情况下执行基于软件的散列表查找以检索所述所请求的散列表值和存储在所述软件可存取存储器中的所述经编码散列表整数密钥的一部分;以及

通过将经编码散列表整数密钥的所述所检索部分经由第二专用硬件指令加载到所述硬件处理器的专用存储器中来更新所述专用硬件散列表实施方案。

50. 根据权利要求 49 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得所述第二专用硬件指令为从所述硬件处理器的所述专用存储器检索值的 GET 指令。

51. 根据权利要求 50 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得所述将散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一动态软件语言操作请求执行所述经编码散列表整数密钥的一部分的加载。

52. 根据权利要求 50 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得:

所述动态软件语言代码为 JavaScript® 代码;且

编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥包括编译 JavaScript 代码以将 JavaScript 散列表字符串密钥编码为散列表整数密钥。

53. 根据权利要求 52 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得将 JavaScript® 散列表字符串密钥编码为散列表整数密钥在编译时间正好实现一次,且依据每一 JavaScript® 操作请求执行所述经编码散列表整数密钥的一部分的加载。

54. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得所述第一专用硬件指令为通过将值存储在存储器映射位置中而将所述值推送到所述专用硬件散列表实施方案的 STORE 指令。

55. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使处理器执行操作使得将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中包括在耦合到主处理器的协处理器上将所述经编码散列表整数密钥的一部分加载到专用硬件散列表实施方案中。

56. 根据权利要求 44 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以可由包括第一处理器和第二处理器的处理器执行,且其中所述所存储的处理器可执行指令经配置以致使所述处理器执行操作使得编译所述动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥是由所述第一处理器执行,且将所述经编码散列表整数密钥的一部分经由第一专用硬件指令加载到专用硬件散列表实施方案中是由所述计算装置的所述第二处理器执行。

57. 根据权利要求 56 所述的非瞬时处理器可读媒体,其中所述第二处理器为专门散列

表协处理器。

58. 根据权利要求 57 所述的非瞬时处理器可读媒体,其中所述所存储的处理器可执行指令经配置以致使所述处理器执行进一步包括以下各项的操作:

对于需要散列表查找的每一动态软件语言操作,通过在所述专门散列表协处理器上执行第二专用硬件指令而确定是否可检索所述散列表值;以及

在确定可检索所述散列表值的情况下在所述专门散列表协处理器上执行基于硬件的散列表查找以接收所述计算装置的所述第一处理器上的所述对应的散列表整数密钥。

对于动态语言的散列表的硬件支持

[0001] 相关申请案

[0002] 本申请案主张 2011 年 5 月 27 日申请的标题为“对于动态语言的散列表的硬件支持 (Hardware Support for Hashtables in Dynamic Languages)”的第 61 / 491,075 号美国临时专利申请案的优先权权益,所述临时专利申请案的全部内容出于所有目的以引用的方式并入本文中。

技术领域

[0003] 本申请案大体涉及计算机软件和硬件,且更特定来说涉及用于在加速散列表操作执行的同时减少执行散列表操作的能量成本的方法、系统和装置。

背景技术

[0004] 动态脚本语言是计算机编程和软件开发中的优选开发平台。特定来说,JavaScript®是针对网页和网络应用的流行的开发语言。随着网页和网络应用的用户需要越来越多的特征,动态脚本语言的复杂性和量持续增加。因此,计算装置解析、解译、分析和执行脚本花费的时间量增加,从而导致网页加载时间变慢且使用户体验降级。

[0005] 例如JavaScript®等某些动态软件语言可视为在其核心处实施散列表功能性。在计算机科学中,散列表或散列映射图是使用散列函数将称为密钥(例如,人名)的识别值映射到其相关联值(例如,其电话号码)的数据结构。因此,散列表实施关联性阵列。散列函数用于将密钥变换为待寻求对应值的阵列元素(或存储器位置)的索引(散列)。

发明内容

[0006] 通过将比如JavaScript®等软件语言的操作表征为基本散列表操作,可使用专门硬件散列表处理器来辅助处理,从而改进计算装置的效率、速度和能量消耗。本文揭示的各个方面提供实施有效散列表的特征的JavaScript®运行时引擎的方法、系统和装置。在一方面中,编译JavaScript®代码以将JavaScript®散列表字符串密钥编码为散列表整数密钥,且可将经编码散列表整数密钥存储在可由软件存取的存储器中。经编码散列表整数密钥的一部分可经由第一专门硬件指令加载到硬件处理器的专门存储器中。对于需要散列表查找的JavaScript®操作,可执行处理器上的专门硬件指令以确定散列表值是否可由硬件处理器检索。如果密钥存储在硬件处理器中,那么可执行基于硬件的散列表查找以传回对应的散列表整数密钥。如果密钥不存储在硬件处理器中,那么可执行基于软件的散列表查找以检索所请求的散列表值,且经编码散列表整数密钥的一部分可存储在软件可存取存储器中。在一方面中,可通过将经编码散列表整数密钥的所检索部分经由第二专门硬件指令加载到硬件处理器的专门存储器中来更新硬件处理器的专门存储器。

[0007] 一方面包含一种用于执行动态软件语言代码的方法,所述方法包含以下操作:编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥;将经编码散列表整数密钥存储在可由软件存取的存储器中;以及将经编码散列表整数密钥的一部分经由第一专

门硬件指令加载到专门硬件散列表实施方案中。在一方面方法中,第一专门硬件指令可为将值推送到专门硬件散列表实施方案的 SET 指令。一方面方法可进一步包含:通过在硬件处理器上执行关于需要散列表查找的每一动态软件语言操作的专门硬件指令来确定散列表值是否可由硬件处理器检索;以及在密钥存储在硬件处理器中的情况下执行基于硬件的散列表查找以传回对应的散列表整数密钥。一方面方法可进一步包含:在密钥不在硬件处理器上的情况下执行基于软件的散列表查找以检索所请求的散列表值和存储在软件可存取存储器中的经编码散列表整数密钥的一部分;以及通过将经编码散列表整数密钥的所检索部分经由第二专门硬件指令加载到硬件处理器的专门存储器中来更新专门硬件散列表实施方案。在一方面方法中,第二专门硬件指令可为从硬件处理器的专门存储器检索值的 GET 指令。在一方面方法中,将散列表字符串密钥编码为散列表整数密钥可在编译时间正好实现一次,且依据每一动态软件语言操作请求执行经编码散列表整数密钥的一部分的加载。在一方面方法中,动态软件语言代码可为 JavaScript® 代码,且编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥可包含编译 JavaScript 代码以将 JavaScript 散列表字符串密钥编码为散列表整数密钥。在一方面方法中,将 JavaScript® 散列表字符串密钥编码为散列表整数密钥可在编译时间正好实现一次,且依据每一 JavaScript® 操作请求执行经编码散列表整数密钥的一部分的加载。在一方面方法中,第一专门硬件指令可为通过将值存储在存储器映射位置中而将值推送到专门硬件散列表实施方案的 STORE 指令。在一方面方法中,将经编码散列表整数密钥的一部分加载到专门硬件散列表实施方案中可包含在耦合到主处理器的协处理器上将经编码散列表整数密钥的一部分加载到专门硬件散列表实施方案中。在一方面方法中,编译动态软件语言代码以将散列表字符串密钥编码为散列表整数密钥可由计算装置的第一处理器执行,且将经编码散列表整数密钥的一部分经由第一专门硬件指令加载到专门硬件散列表实施方案中由计算装置的第二处理器执行。在一方面方法中,计算装置的第二处理器可为专门散列表协处理器,且所述方法可进一步包含:通过在专门散列表协处理器上执行关于需要散列表查找的每一动态软件语言操作的第二专门硬件指令而确定是否可检索散列表值;以及在确定可检索散列表值的情况下在专门散列表协处理器上执行基于硬件的散列表查找以接收计算装置的第一处理器上的对应的散列表整数密钥。

[0008] 另一方面包含一种计算装置,所述计算装置包含耦合到存储器的处理器,其中处理器配置有处理器可执行指令以执行上文描述的各个方面方法的操作。另一方面包含计算装置,所述计算装置包含用于执行上文描述的各个方面方法的功能和操作的装置。另一方面包含非瞬时处理器可读媒体,所述非瞬时处理器可读媒体上存储有处理器可执行指令,所述处理器可执行指令经配置以致使处理器执行上文描述的各个方面方法的操作。

附图说明

[0009] 并入本文且构成本说明书的部分的附图说明本发明的示范性方面,且连同上文给出的一般描述和下文给出的详细描述一起用以阐释本发明的特征。

[0010] 图 1 是说明实例阵列的特征的数据结构图。

[0011] 图 2 是说明关联性阵列的实例实施方案的特征的数据结构图。

[0012] 图 3 是说明关联性阵列的散列表实施方案的特征的数据结构图。

- [0013] 图 4 是说明实例页表翻译系统的处理器的功能组件图。
- [0014] 图 5A 是根据各个方面用于在硬件中执行散列表操作的计算装置架构的功能组件图。
- [0015] 图 5B-5D 是根据各个方面用于在硬件中执行散列表操作的包含存储器和处理器组件的替代计算装置架构的电路框图。
- [0016] 图 6 是适于与各个方面一起使用的实例方法的过程流程图。
- [0017] 图 7 是适于与各个方面一起使用的移动计算装置的切开透视图。
- [0018] 图 8 是适于与各个方面一起使用的实例计算机的说明。

具体实施方式

[0019] 将参看附图详细描述各个方面。只要可能,将在整个图式中使用相同的参考标号来指代相同或相似的部分。对具体实例和实施方案做出的参考是出于说明性目的,且不希望限制本发明或权利要求书的范围。

[0020] 本文使用词语“示范性”来表示“充当实例、例子或说明”。本文描述为“示范性”的任何实施方案均不一定解释为与其它实施方案相比是优选或有利的。

[0021] 术语“电子装置”、“移动装置”和“计算装置”可在本文互换使用以指代个人计算机、蜂窝式电话、个人数据助理 (PDA)、掌上型计算机、平板计算机、无线电子邮件接收器 (例如, **Blackberry®** 装置)、具有多媒体因特网功能的蜂窝式电话,以及包含可编程处理器的类似个人电子装置中的任一者或全部。虽然各个方面在可具有有限处理能力的移动装置 (例如,蜂窝式电话) 中尤其有用,但所述方面通常可用于执行以动态、脚本撰写和 / 或标记语言编写的脚本和 / 或应用的任何计算装置。

[0022] 术语“动态语言”和“脚本语言”在本申请案中一般地且可互换地使用以指代任何动态语言、脚本语言,或指代用于编写在运行时解译和 / 或编译的程序 (本文为“脚本”) 的任何所解译语言。这些术语还可指代在所管理运行时运行的任何语言。因此,出于本申请案的目的,术语“动态语言”和“脚本语言”不应限于从源代码或字节码解译的语言,也不应限于连同传统上编译为本机代码的程序一起执行的语言。本申请案范围内的动态和脚本语言的实例包含 (例如) **JavaScript®**、Perl、Python 和 Ruby,以及将来可能开发的 Java 和其它语言。

[0023] 为便于参考,使用 **JavaScript®** 术语描述各个方面。然而,除非权利要求书中明确叙述,否则此术语的使用以及 **JavaScript®** 实例的使用不应解释为将权利要求书的范围限于特定软件语言或技术。

[0024] 图 1-3 说明各个已知数据结构的数据结构元件和基本操作。由于这些结构是众所周知的,所以已省略许多细节 (例如,不同类型数据的存储、存储器大小等) 以便集中于对相关特征的描述。

[0025] 图 1 是存储雇员识别符 (ID) 值的实例阵列 100 的说明。阵列通常将若干值循序存储在存储器中,使得每一值可为通过其在存储器中的相对位置的参考 (即,索引)。举例来说,图 1 说明实例阵列 100 循序存储雇员 ID 值,每一值 (例如, ID) 与一个索引 (例如, 1、2、3) 相关联。阵列的循序性质意味着基于索引从阵列检索值的指令 (例如, `getItem[2]`) 通常要求系统存取阵列中的第一元件的存储器位置 (例如,存储器位置

“1005”),且将存储器位置递增索引值(例如,“2”)以定位所要项目(例如,存储器位置“1007”处的值“489954”)。阵列内的特定值的位置通常是阵列未知的。为找到特定值,可检查阵列中的每一项目直到找到所述值为止,且识别与所述值相关联的索引。

[0026] 图2是关联性阵列200的实例实施方案的说明。关联性阵列是由唯一密钥集合和值集合组成的抽象数据类型,其中每一密钥与一个值或值集合相关联。图2说明关联性阵列200如何含有识别可存储所要值(例如,雇员ID)的其它存储器位置(例如,存储器位置20、105和155)的唯一密钥值(例如,整数值20、105和155)。图2还说明可相对于例如雇员姓名204、电话号码206、地址208和职位210等其它相关信息存储每一雇员ID202。以此方式,当定位所要值(例如,雇员ID)时,还可基于所要值在存储器中相对于所要值的位置来定位与所要值相关的其它信息。举例来说,基于值的索引检索值的指令(例如,getItem[2])可指示处理器存取阵列中的第一元件的存储器位置(例如,存储器位置1005),将所述存储器位置递增索引值(例如,2)以定位所要密钥值(例如,存储器位置1007处的密钥155),存取与密钥值相关联的第二存储器位置(例如,存储器位置155),且传回存储在第二存储器中的值(或到所述值的指针)(例如,存储器位置155处的值489954)。因此,与所要值相关的信息(其相对于所要值存储)可与所要值同时检索。

[0027] 虽然上文参看图2论述的实例实施方案允许多个项目与单一阵列元件(即,密钥)相关联,但存取阵列元件仍是循序过程(例如,索引值仍为偏移)。在此类型的数据结构中,无法用高效的方式在不遍历、搜索和/或分类阵列的情况下确定给定值的位置(例如,索引)。虽然存在用于分类和/或搜索阵列的许多技术,但阵列操作仍相对昂贵且缓慢。

[0028] 图3说明关联性阵列的实例散列表实施方案。散列表使用散列函数以通过将密钥值变换为待寻求对应值的阵列元件的索引(例如,存储器位置)而将密钥映射到其相关联值。图3说明其中阵列存储器位置(例如,存储器位置1005、1006、1007)可含有雇员ID作为密钥值的实例。可对密钥值(例如,雇员id)执行散列函数以产生识别对应值的位置(例如,存储器位置739、846、954)的散列值(例如,846、739、954)。散列函数可使用大量可用数学公式的任一者将密钥值转换为散列值。在图3中说明的实例中,散列函数使用密钥值的最后三位作为散列值且存储相对于散列识别的位置的相关信息(例如,姓名)。以此方式,从散列表检索值的指令(例如,getItem[489954])简单地要求对密钥值(例如,489954)进行散列操作,且存取散列值(954)的存储器位置以定位所要项目(例如,存储器位置954处的John Doe)。

[0029] 因为散列表非常快速且高效,所以实施基本散列表功能性大大改进了效率、速度、能量消耗,以及许多其它性能因素。然而,散列表通常为软件结构,且现代硬件制造商不提供提供本机散列表功能性的处理器。

[0030] 各个方面提供用于允许应用在硬件中执行散列表操作(例如,查找)的方法、系统和装置。散列表操作的执行可由微处理器上的特殊指令寄存器经由存储器映射输入/输出(MMIO)加速器装置和/或通过使用一个或一个以上协处理器来辅助或控制。各个方面还将散列表功能性与局部性原理组合以大大改进实施基本散列表功能性的软件语言(例如,JavaScript®)的性能。

[0031] “局部性原理”是通常与设计虚拟存储器 and 高速缓冲存储器相关联的原理。局部性原理通常陈述,如果一个过程请求存取存储器中的项目,那么极有可能在不久的将来某一

时间请求存取接近所述存储器的另一项目。根据所述原理,获取其中定位所述项目的整个存储器块(与仅检索所请求项目相比)且将所获取块移动到更快且更可评估存储器中以供将来存取通常是有益的。

[0032] 硬件系统实施利用局部性原理的技术是有益的。如今的微处理器在比现代存储器快得多的速度下操作。因此,处理过程中的主要硬件瓶颈是其存取外部存储器所花费的时间。举例来说,归因于存储器的相对缓慢的速度以及因此实现存储器存取所需的时间,微处理器可进入涉及存储器读取或写入操作的处理器操作之间的等待状态(或将处理交给不同线程)以便允许存储器有足够的时间准备好进行下一操作。为缩短此处理器等待时间,现代计算机架构使用使存储器存取的速度与处理器的操作速度匹配的高速缓冲存储器。高速缓冲存储器是处理器可比计算装置中的其它存储器(例如,硬盘、基于电容器的存储器、随机存取存储器、只读存储器等)快得多地存取的小型、昂贵且快速的存储器。每当请求给定存储器地址处的字节时,处理器首先试图从较快高速缓冲存储器获得数据。然而,如果高速缓冲存储器不含有所请求数据,那么处理器进入等待状态,此时从其它较慢系统存储器的一者加载所请求数据。因此,高速缓冲存储器的效力(例如,处理速度的总体改进)与高速缓冲存储器含有处理器请求的数据的频率直接相关。

[0033] 为使性能最大化,处理器或高速缓冲存储器控制器可预测处理器接下来或在将来操作中即将需要的数据,且在处理器请求所述数据之前将所述数据加载到高速缓冲存储器中。由于定位给定存储器位置比从存储器读取数据慢得多,所以现代高速缓冲存储器利用局部性原理(即,处理器趋向于在给定时间点仅存取地址空间的小部分)来加载整个数据块,预期存储在所参考存储器附近的数据也可能在不久的将来被处理器参考。

[0034] 除了利用局部性原理改进总体处理器性能外,各个操作系统的现代页表翻译系统还使用局部性原理将页表从虚拟存储器翻译到物理存储器。因此,操作系统可同时运行比原本将纳入物理存储器中更多的过程。

[0035] 图4说明实施实例页表翻译系统400的计算装置的功能组件。如图4中说明,在计算装置上运行的操作系统402可利用物理存储器406与虚拟存储器408的组合来支持大量过程410的同时执行。物理存储器406提供受实际上安装在计算机中的存储器模块的大小和数目限制的地址空间(存储器1-n)。相比之下,虚拟存储器408提供允许计算装置同时运行比实际上纳入物理存储器406中多的过程的大得多的地址空间(存储器1-m)。即,虚拟存储器408的存在允许计算装置主要从较快但较小的存储器(例如,物理存储器406)工作,同时能够可以存取存储在较慢但大得多的大容量存储器(例如,虚拟存储器408)中的信息。每当过程请求存取不在物理存储器406中的信息时,计算装置可从虚拟存储器408获取信息且将所请求信息(或含有信息的块)移动到物理存储器406。页表412可用于将物理存储器406映射到虚拟存储器408。下文更详细论述页表412。

[0036] 图4说明操作系统402可包含页表412,页表412存储用于实施虚拟存储器408与物理存储器406之间的地址翻译方案(即,地址映射)的映射信息。同样,硬件404可包含含有来自页表412的最近使用的映射的翻译查找缓冲器414。每当需要虚拟到物理地址翻译时,系统可首先检查翻译查找缓冲器414以确定缓冲器414是否含有必要的映射信息。如果翻译查找缓冲器414含有必要的映射信息(例如,找到匹配),那么将物理存储器地址提供到发出请求的过程,且数据存取继续进行。然而,如果未找到匹配,那么操作系统402

检查页表 412 以确定页表 412 是否含有所请求的映射信息。如果在页表 412 中找到映射信息,那么可将信息(或其含有信息的块)写入到硬件 404 的翻译缓冲器 414,其后数据存取继续进行。

[0037] 如上文论述,页表 412 可将物理存储器 406 映射到虚拟存储器 408,且翻译查找缓冲器 414 可含有最近使用和/或预测的物理存储器映射信息的块。依据映射的性质,可将翻译查找缓冲器 414 视为硬件散列表的原始硬件实施方案。然而,即使当就此方面考虑时,当前解决方案(例如,页表翻译系统)也在处理器的高度保护层中实施此映射,使得其仅在“超级用户模式”中、仅对于操作系统过程,且仅针对一种类型的应用(例如,页表查找)是可存取的。

[0038] 各个方面提供包含专门处理器电路的系统、方法和装置,所述专门处理器电路实施可供应用使用的基于硬件的散列操作。各个方面准许对硬件散列表操作的用户层级(例如,用户模式)许可/存取。各个方面提供产生代码的编译器,所述代码可在软件和硬件散列表逻辑两者中解译和/或执行。各个方面使应用开发者能够利用例如JavaScript®等各种软件语言的基本结构,来改进软件程序和过程的效率和执行/解译。

[0039] 如上文提及,比如JavaScript®等某些软件语言可视为在其核心处实施散列表功能性。JavaScript®是其中最简单结构为“对象”数据类型的对象定向脚本语言。在JavaScript®中,对象实施为指定特性的集合,所述指定特性简单地为另一JavaScript®对象拥有的JavaScript®对象(即,其是另一对象本地的对象)。特性可使用点运算符(例如,“object.property”)或括弧(例如,“object[property]”)来存取。考虑到散列表使用类似语法和逻辑定位/存取元件,每一JavaScript®对象可视为散列表实施方案。举例来说,在JavaScript®中,向对象“o”的特性“x”写入(例如,o.x=10)可视为等效于对象“o”中的散列插入(例如,o[x]=10),且读取对象“o”的特性“x”(例如,z=o.x)可视为等效于对象“o”中的散列读取(例如,z=o[x])。当就此方面考虑时,每个JavaScript®操作可表征为从字符串值(密钥)经过另一字符串或数字值(值)的特殊类型的散列操作,且每个JavaScript®对象可视为关联性阵列的散列表实施方案。

[0040] 通过将每一JavaScript®操作表征为基本上是散列表操作,可使用专门散列表处理器来处理JavaScript®对象。专门散列表处理器可包含经保留用于执行散列表操作的电路(内嵌在微处理器中)。处理器可配置有特殊指令寄存器,其允许应用存取硬件散列表逻辑以执行JavaScript®功能性。特殊指令寄存器可使用经修改指令集架构(ISA)存取,所述ISA允许软件应用以减少数目的指令存取基于硬件的散列操作。举例来说,可将“get”机器语言指令和“set”机器语言指令添加到专门散列表处理器,使得JavaScript®编译器和应用可更容易地使用单一指令(例如,GET或SET)在硬件中执行散列表查找。还可经由正常ISA与特殊寄存器或存储器地址的组合来存取散列表的硬件实施方案。

[0041] 图 5A 说明用于在硬件与软件的组合中执行散列表操作(例如,JavaScript)的实例计算装置架构 500。图 5A 说明计算装置架构可包含与硬件 504 通信的操作系统 502,硬件 504 具有专门处理器 510 和一个或一个以上硬件散列表 518。专门处理器 510 可包含能够执行例如将值读取和写入到硬件散列表 518 等散列表操作的专门电路。处理器 510 可进一步经配置以包含“get”机器语言指令寄存器 506 和“set”机器语言指令寄存器 508,其一起提供对专门电路和硬件散列表 518 的存取。

[0042] 图 5B 说明用于在硬件与软件的组合中执行散列表操作的硬件 504 的实例实施例硬件架构 530。此实施例硬件架构 530 可包含中央处理单元 520, 其包含用于执行算术和逻辑运算的算术逻辑单元 (ALU) 522、用于执行浮点运算 (例如, 加法、减法、乘法、除法、平方根等) 的浮点单元 (FPU) 524, 以及硬件散列表实施单元 528。硬件散列表实施单元 528 可包含对应于上文论述的硬件散列表 518 的用于存储散列表值的寄存器 / 存储器。这些寄存器和 / 或存储器有时在本文中称为散列表存储器。

[0043] 图 5C 说明用于在硬件与软件的组合中执行散列表操作的硬件 504 的另一实例实施例硬件架构 560。此实施例硬件架构 560 可包含中央处理单元 (CPU) 520, 其包含算术逻辑单元 (ALU) 522 和浮点单元 (FPU) 524, 类似于上文论述的架构 530。硬件架构 560 还可包含与 CPU520 分离的硬件散列表实施单元 528。举例来说, 硬件散列表实施单元 528 可在专门协处理器 (未说明) 中, 所述协处理器执行散列操作且将结果经由并行或串行输入 / 输出端口传送到 CPU520。

[0044] 图 5D 说明用于在硬件与软件的组合中执行散列表操作的另一实例实施例硬件架构 590。此实施例硬件架构 590 可包含中央处理单元 (CPU) 520, 其包含算术逻辑单元 (ALU) 522 和浮点单元 (FPU) 524, 类似于上文论述的架构 530、560。硬件架构 560 还可包含与 CPU520 分离的硬件散列表实施方案 528 和存储器映射输入 / 输出 (MMIO) 加速器装置 592。硬件架构 590 可利用 MMIO 加速器装置 592 在 CPU520 外部执行散列操作且将结果经由处理器总线传送到 CPU520。

[0045] 返回到图 5A, 操作系统 502 可包含具有编译器散列模块 516 的编译器 512, 编译器散列模块 516 在编译时间将 JavaScript® 字符串映射 (例如, 散列操作) 到整数值。在一方面中, 此映射仅进行一次。在一方面中, 所述映射可用于产生 JavaScript® 对象的软件散列表 514 表示。此映射很重要, 因为用于处理散列表的传统方法不能用于在硬件中处理 JavaScript® 对象。如上文论述, JavaScript® 对象可视为将字符串值 (密钥) 链接到另一字符串或数字值 (值) 的特殊类型的散列表。将字符串值用作密钥 (而非整数值) 使对来自散列表的值进行散列操作、存储和检索的过程变复杂。举例来说, 字符串可含有任意数目的字符 (即, 其可为任何长度), 而整数密钥值含有已知数目的位或字节。将字符串用作密钥值排除使用简单的数学算法来产生散列值, 例如使用上文参看图 3 论述的密钥的最后几位。

[0046] 出于这些和其它原因, 对具有字符串值作为密钥的散列表执行散列表操作需要执行数千个软件指令。当对 JavaScript® 对象执行这些散列操作时尤其如此, 其中嵌套散列 (JavaScript® 的特征) 的存在可将单一散列表查找请求转变为数千散列表查找请求, 其每一者可能需要执行数千个软件指令。

[0047] 用于使用字符串作为散列表密钥的当前解决方案需要无法在硬件中有效实施的较长且复杂的基于软件的处理。为克服这些限制, 编译器散列模块 516 可在编译时间 (即, 与运行时对比) 以允许由硬件组件执行散列表操作 (例如, JavaScript® 代码) 的方式将 JavaScript® 字符串映射到整数值。编译器散列模块 516 可产生软件散列表 514, 其是多个散列表 (例如, JavaScript® 对象) 中的一个散列表 (例如, 字符串 - 整数映射)。此双散列操作 (或嵌套散列表) 允许将软件散列表 514 的一些部分存储在硬件散列表 518 中, 使得所存储的信息可使用单一读取或单一写入操作来存取。这将执行来自通常需要的数千个散

列操作的每一散列操作所需的指令数目减少为单一指令（例如，读取或写入）。

[0048] 如上文提及，每个JavaScript®对象可视为散列表，且一般化散列表处理器（实施在硬件中）可用于处理JavaScript®代码。举例来说，每一JavaScript®对象可在处理器 510 的一般化散列表电路中以比当前解决方案明显少的指令处理。此减少数目的所需指令大大改进JavaScript®应用的总体速度和效率。

[0049] 返回图 5，编译器散列模块 516 可仅执行用于字符串 - 整数映射的过程一次，这允许由硬件以单一指令执行后续散列表更新和查找。此映射还允许将传统上内嵌在实时编译器代码中的一些散列表支持移动到指令集架构 (ISA) 扩展部分，例如 GET 指令和 SET 指令。

[0050] 在一方面中，处理器可具有用于处理 GET 和 SET 指令的三个寄存器，其可以用三个寄存器参数执行。前两个参数（例如，r1、r2）可用于计算密钥，且第三参数可用于复制 GET 指令的所得值且用于存储源自 SET 指令的值。以下伪码说明三个参数的使用：

[0051] `get r1, r2, r3 // r3=HASH[key(r1, r2)]`

[0052] `set r1, r2, r3 // HASH[key(r1, r2)]=r3`

[0053] 在以上伪码中，前两个寄存器（对应于参数 r1 和 r2）是源寄存器，且第三寄存器（对应于参数 r3）是目的地寄存器。将两个寄存器（r1、r2）用作密钥有助于在对象间分离条目且辅助处理嵌套散列，且因此非常适于例如JavaScript®等语言。举例来说，在JavaScript®的情况下，密钥的一个分量（例如，r1）可映射到JavaScript®对象，且密钥（例如，r2）的另一分量可映射到JavaScript®对象的特性。“密钥”表示由 r1 和 r2 构造散列表密钥的数学变换。“散列”是基于所计算密钥执行值查找的实际散列表实施方案。

[0054] 在一方面中，可使指令寄存器 506、508 可由操作系统在用户模式中存取（与页表翻译所需的超级用户模式对比）。这允许在操作系统上执行的任何过程或应用在用户模式中操作的同时发布上文提及的 GET 和 SET 指令。在一方面中，硬件散列表 516 可高速缓存散列表值的块，使得基于硬件的散列功能性的组合在用户模式中可供如下文更充分论述的每一过程使用。

[0055] 每当软件过程请求散列操作（例如，查找）时，计算装置处理器可首先检查硬件（即，高速缓冲存储器）以经由 GET 指令查看所请求散列值的存在。如果所述值在硬件（即，高速缓冲存储器）中，那么指令寄存器可简单地传回所请求的值，从而允许在一个指令中完成散列操作。如果 GET 指令指示所请求项目不在硬件散列表 514 中，那么可作出对软件散列表 514 查找过程的调用。查找过程可在软件中（例如，在软件散列表 514 上）执行所请求的散列操作，且传回将来可能存取的所请求的值和周围值两者。这些值可推送到硬件高速缓冲存储器 504 且存储在硬件散列表 516 中（例如，经由 SET 指令），使得其容易由处理器 510 使用用于将来处理。以此方式，各个方面通过将所产生的散列信息块经由 SET 指令（例如，在“set”寄存器 508 上发布）推送到处理器 510 高速缓冲存储器而利用局部性原理。此操作使得能够将将来可能存取的值存储在硬件散列表 514 高速缓冲存储器中以供后续操作中的处理器快速存取（例如，经由单一指令）。

[0056] 为产生软件散列表 514，编译器 512 可在代码编译期间产生将唯一 ID 指派到每个字符串字面值的全局字符串表。对于动态对象，代替于具有高速缓冲存储器，编译器可使用保持JavaScript®特性的值的阵列。为存取特性，编译器可查找特性名称的字符串 ID 且使用另一散列表，所述另一散列表映射对象 ID 与特性字符串 ID 的组合且将所述组合映射到整

数,所述整数表示指示特性值的对象中的字段阵列索引。以下伪码说明此过程:

```
[0057] // z=o.x
[0058] ID1=StringtoIDHash[“x”]; // 第一散列查找
[0059] ID2=o.ID
[0060] ID3=IdToIndexHash[ID1, ID2] // 第二散列查找
[0061] // 使用 GET 指令
[0062] Z=o.Fields[ID3]
```

[0063] 在以上伪码中,第一散列查找仅发生一次且在编译时间发生。另一方面,第二散列查找可每当执行代码时发生。然而,由于第二散列查找使用 GET 指令,所以仅执行一个指令,条件是所请求的散列信息在硬件散列表中。

[0064] 如上文论述,可将散列密钥经由 SET 指令推送到硬件,使得将来最可能存取的值可存储在硬件(即,高速缓冲存储器)中。以下伪码说明 SET 指令的使用:

```
[0065] // o.x=z
[0066] ID1=StringtoIDHash[“x”];
[0067] ID2=o.ID
[0068] ID3=IdToIndexHash[ID1, ID2] // 使用 GET 指令
[0069] IF (ID3== -1) {
[0070] ID3=o.Fields.Legth
[0071] IdToIndexHash[ID1, ID2]=ID3; // 使用 SET 指令
[0072] }
[0073] o.Fields[ID3]=z
```

[0074] 图 6 说明用于在硬件与软件的组合中执行散列表操作(例如,JavaScript®操作)的方面方法 600。在方法 600 中框 602 处,编译器可通过将唯一 ID 指派到每个字符串字面值(例如,JavaScript®对象)而将JavaScript®字符串密钥映射到整数,以在编译时间产生主控散列表(其存储在软件中)。在框 604 中,可将具有散列表电路的硬件处理器的专门指令寄存器(ISA 扩展部分)初始化,使得其可在用户模式中由软件使用。在确定框 606 中,处理器可等待一个或一个以上软件过程请求散列表操作(例如,查找)。如果软件过程请求散列操作(即,确定框 608=“是”),那么可在框 608 中在专门指令寄存器上向处理器发布 GET 指令。在框 610 中,可使用散列表电路来存取硬件散列表以确定对应于所请求散列表操作的值是否在硬件散列表中。在确定框 612 中,处理器可确定硬件散列表是否含有所请求的值。如果所请求的值在硬件散列表中(即,确定框 612=“是”),那么专门指令寄存器可传回所请求的值,使得散列表操作由硬件在框 614 中执行。此操作可仅使用一个指令执行,且结果可提供到软件过程。如果所请求的值不在硬件散列表中(即,确定框 612=“否”),那么在框 616 中硬件可通知软件过程所述值不可用。在框 616 中,软件过程可请求来自主控散列表的值,使得散列表操作由软件执行。在框 618 中,软件过程可在专门指令寄存器上发布 SET 指令以将含有所请求值的散列表值的块推送到硬件散列表。一旦在框 614 或 618 中传回所请求值,处理器就可返回到确定框 606 以等待一个或一个以上软件过程请求散列表操作(例如,查找)。

[0075] 图 7 是适于与各方面的任一者一起使用的计算装置的系统框图。典型的计算装置

700 可包含耦合到内部大容量存储器 702(例如,快闪存储器)、显示器 703 且耦合到扬声器 764 的处理器 701。另外,计算装置 700 还可包含用于发送和接收电磁辐射的天线 704,其可连接到无线数据链路和 / 或耦合到处理器 701 的蜂窝式电话收发器 705。用于执行硬件中的散列表操作的各个方面方法可由处理器 701 和存储器 702 的部分执行。或者,计算装置 700 内或耦合到计算装置 700 的专门模块可执行所述方面方法。

[0076] 上文描述的方面可在多种硬件配置中实施,包含利用存储器映射输入 / 输出(MMIO) 加速器装置和协处理器的配置。举例来说,MMIO 装置可执行对应于上文论述的 GET 和 SET 指令的 LOAD 和 STORE 操作。在各个方面中,可使用硬件辅助的虚拟机架构来实施各个特征。

[0077] 上文描述的方面还可在多种计算装置中实施,例如图 8 中说明的膝上型计算机 800。膝上型计算机 800 将通常包含耦合到易失性存储器 802 和大容量非易失性存储器(例如,磁盘驱动器 803) 的处理器 801。计算机 800 还可包含耦合到处理器 801 的软盘驱动器 804 和压缩光盘(CD) 驱动器 805。计算装置 800 还可包含耦合到处理器 801 的若干连接器端口,其用于建立数据连接或接纳外部存储器装置,例如 USB 或 FireWire® 连接器插口,或用于将处理器 801 耦合到网络的其它网络连接电路 806。在笔记本配置中,计算机外壳包含触摸板 807、键盘 808 和显示器 809,其全部耦合到处理器 801。计算装置的其它配置可包含如众所周知耦合到处理器(例如,经由 USB 输入)的计算机鼠标或轨迹球。

[0078] 处理器 701、801 可为可由软件指令(应用)配置以执行多种功能(包含下文描述的各个方面的功能)的任何可编程微处理器、微计算机或多个处理器芯片。在一些移动接收器装置中,可提供多个处理器 701,例如一个处理器专用于无线通信功能且一个处理器专用于运行其它应用。通常,软件应用可在其被存取并加载到处理器 701、801 中之前存储在内部存储器 702、802、803 中。在各个方面中,处理器 701、801 还包含足以存储应用软件指令的内部存储器以及用于存储散列表或散列表值的块的高速缓冲存储器。

[0079] 前述方法描述和过程流程图仅作为说明性实例而提供,且不希望需要或暗示必须以所呈现的次序执行各种方面的步骤。如所属领域的技术人员将了解,可以任何次序执行以上方面中的步骤的次序。例如“随后”、“接着”、“接下来”等词语不希望限制步骤的次序;这些词语仅用于引导读者通读对方法的描述。此外,以单数形式(例如,使用冠词“一”或“所述”)对权利要求元件的任何参考不应解释为将所述元件限于单数形式。

[0080] 结合本文中所揭示的方面而描述的各种说明性逻辑块、模块、电路和算法步骤可被实施为电子硬件、计算机软件或两者的组合。为清楚说明硬件与软件的此互换性,上文已大致关于其功能性而描述了各种说明性组件、块、模块、电路和步骤。所述功能性是实施为硬件还是软件取决于特定应用及施加于整个系统的设计约束。所属领域的技术人员可针对每一特定应用以各种方式实施所描述的功能性,但此类实施决策不应被解释为脱离权利要求书的范围。

[0081] 用于实施结合本文揭示的方面描述的各种说明性逻辑、逻辑块、模块和电路的硬件可以用通用处理器、数字信号处理器(DSP)、多媒体广播接收器芯片内的 DSP、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它可编程逻辑装置、离散门或晶体管逻辑、离散硬件组件,或经设计以执行本文描述的功能的其任何组合来实施或执行。通用处理器可以是微处理器,但在替代方案中,处理器可以是任何常规处理器、控制器、微控制器或状态

机。处理器还可实施为计算装置的组合,例如,DSP与微处理器的组合、多个微处理器的组合、一个或一个以上微处理器与DSP核心的联合,或任何其它此配置。或者,一些步骤或方法可由特定用于给定功能的电路执行。

[0082] 在一个或一个以上示范性方面中,可以用硬件、软件、固件或其任何组合来实施所述的功能。如果用软件实施,则可将功能作为计算机可读媒体上的一个或一个以上指令或代码而加以存储或传输。本文揭示的方法或算法的步骤可体现在所执行的处理器可执行软件模块中,所述处理器可执行软件模块可驻留在非瞬时计算机可读或处理器可读媒体上。非瞬时计算机可读媒体包含可由计算机或计算装置的处理器存取的任何物理(即,非瞬时)存储媒体。借助实例而非限制,此类非瞬时计算机可读媒体可包含RAM、ROM、EEPROM、CD-ROM,或其它光盘存储装置、磁盘存储装置或其它磁性存储装置,或可用于携载或存储指令或数据结构的形式所要程序代码且可由计算机或计算装置的处理器存取的任何其它媒体。如本文中所使用,磁盘和光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软磁盘和蓝光光盘,其中磁盘通常磁性地复制数据,而光盘使用激光光学地复制数据。以上各项的组合也应包含在非瞬时计算机可读媒体的范围内。另外,方法或算法的操作可作为存储在非瞬时处理器可读媒体和/或计算机可读媒体上的代码和/或指令的一者或任何组合或集合而驻留,所述非瞬时处理器可读媒体和/或计算机可读媒体可并入到计算机程序产品中。

[0083] 提供先前对所揭示方面的描述以使得所属领域的技术人员能够制作或使用本发明。所属领域的技术人员将容易明白对这些方面的各种修改,且本文中所定义的一般原理可在不偏离本发明的精神或范围的情况下应用于其它方面。因此,本发明不希望限于本文所示的方面,而是应被赋予与所附权利要求书以及本文揭示的原理和新颖特征一致的最广范围。

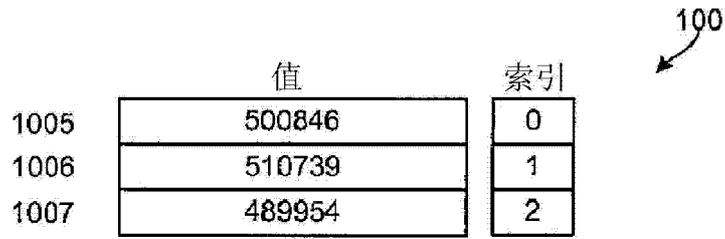


图 1

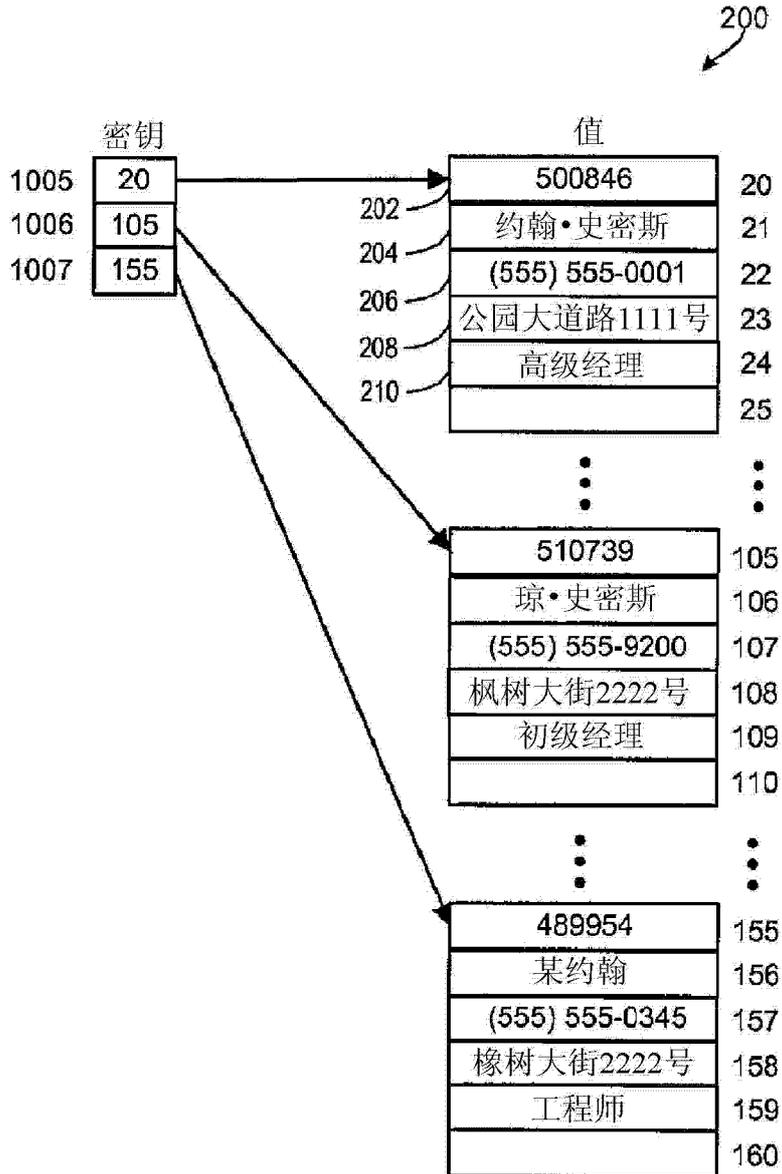


图 2

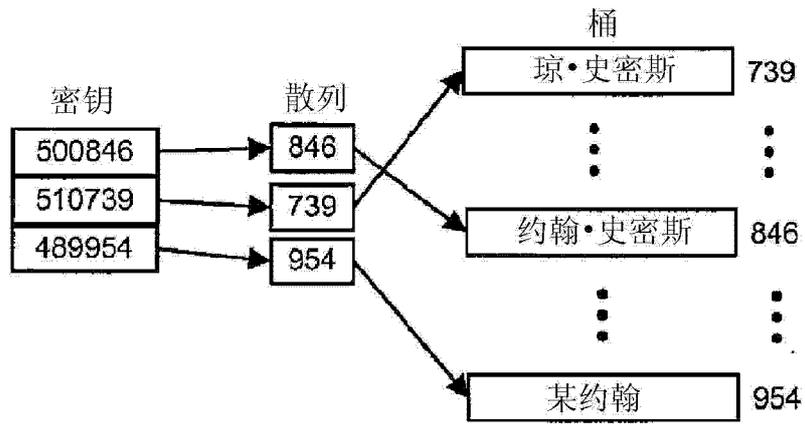


图 3

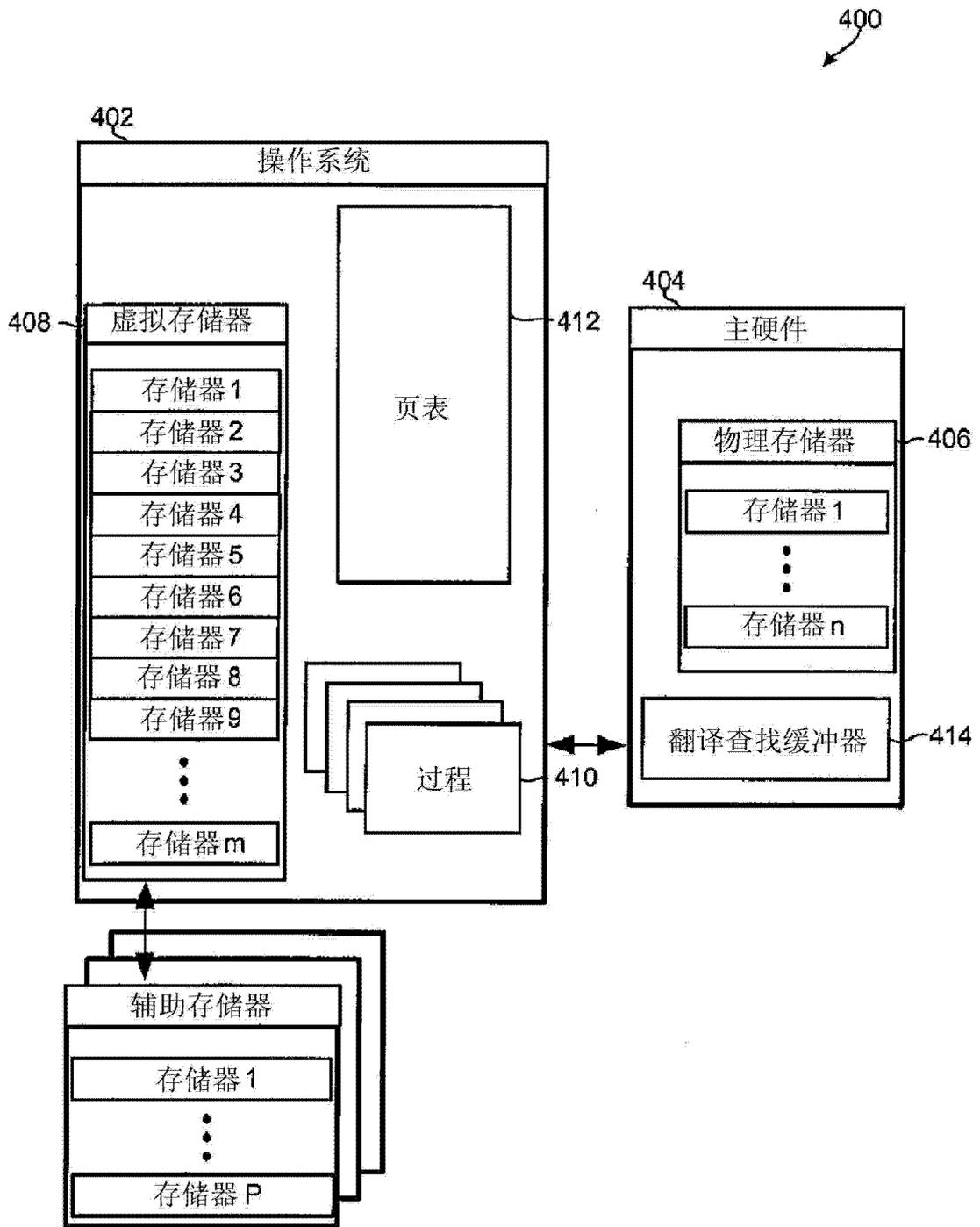


图 4

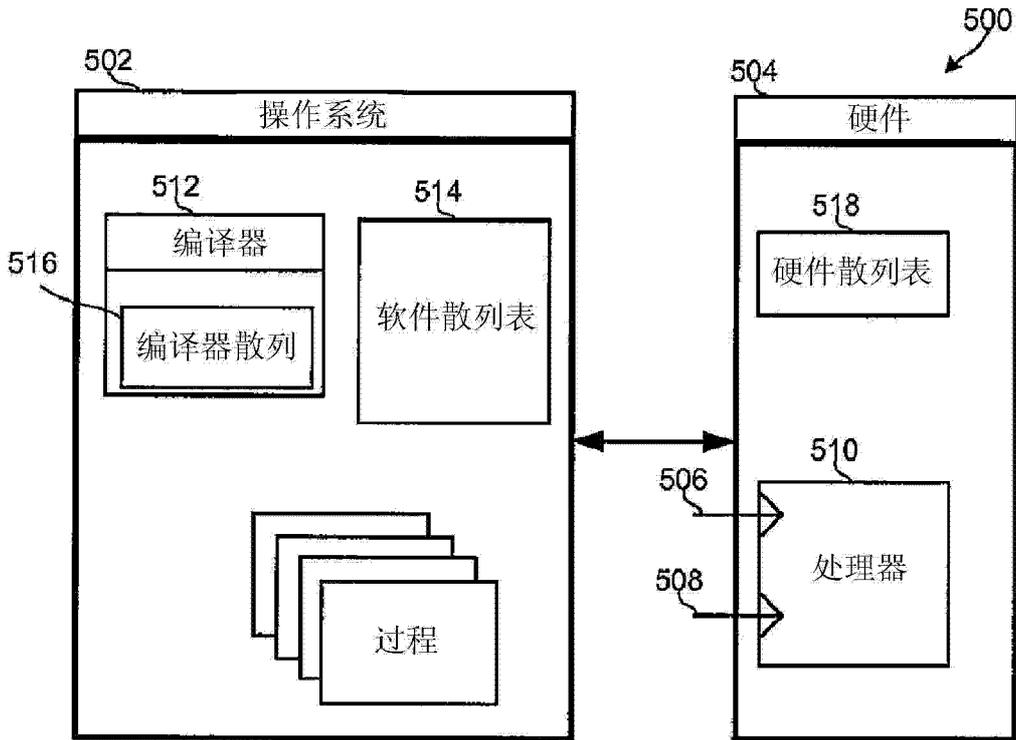


图 5A

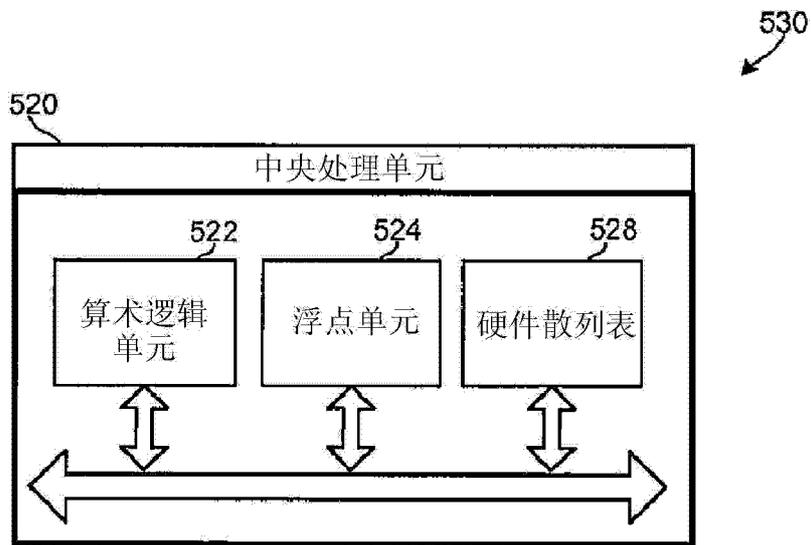


图 5B

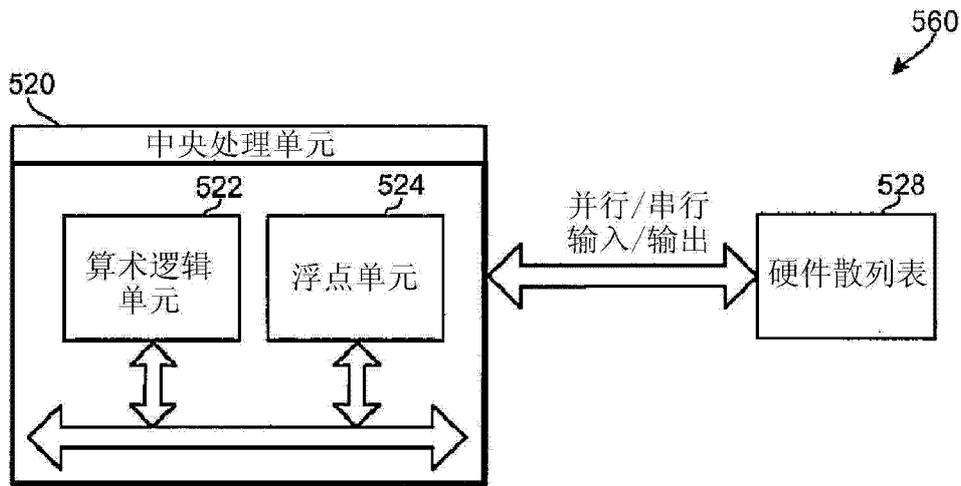


图 5C

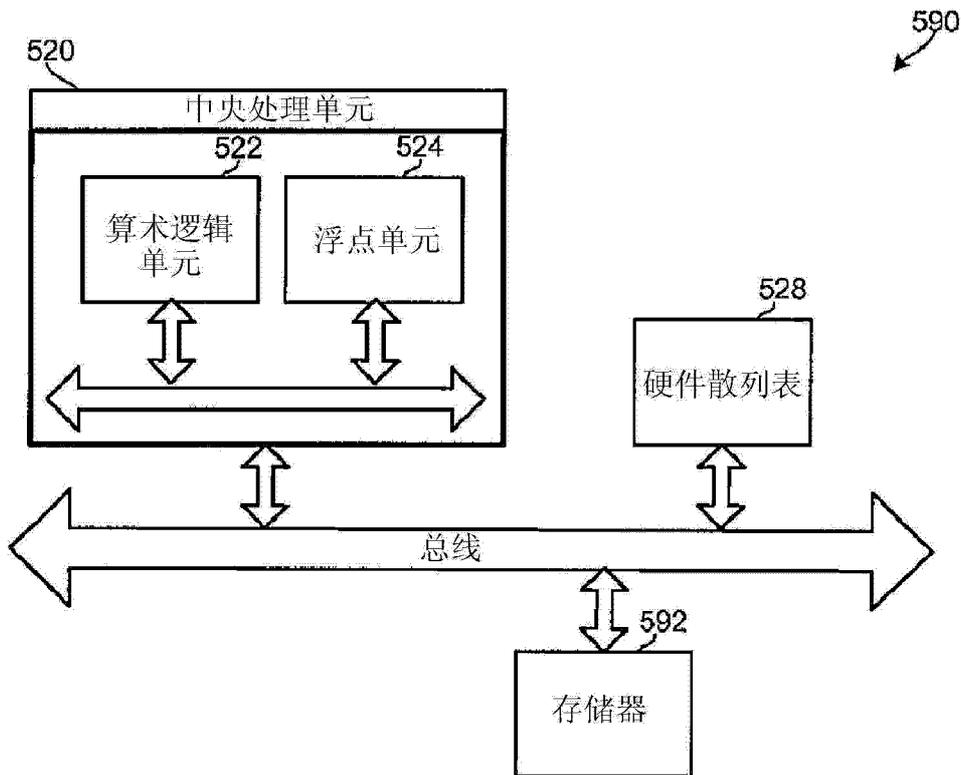


图 5D

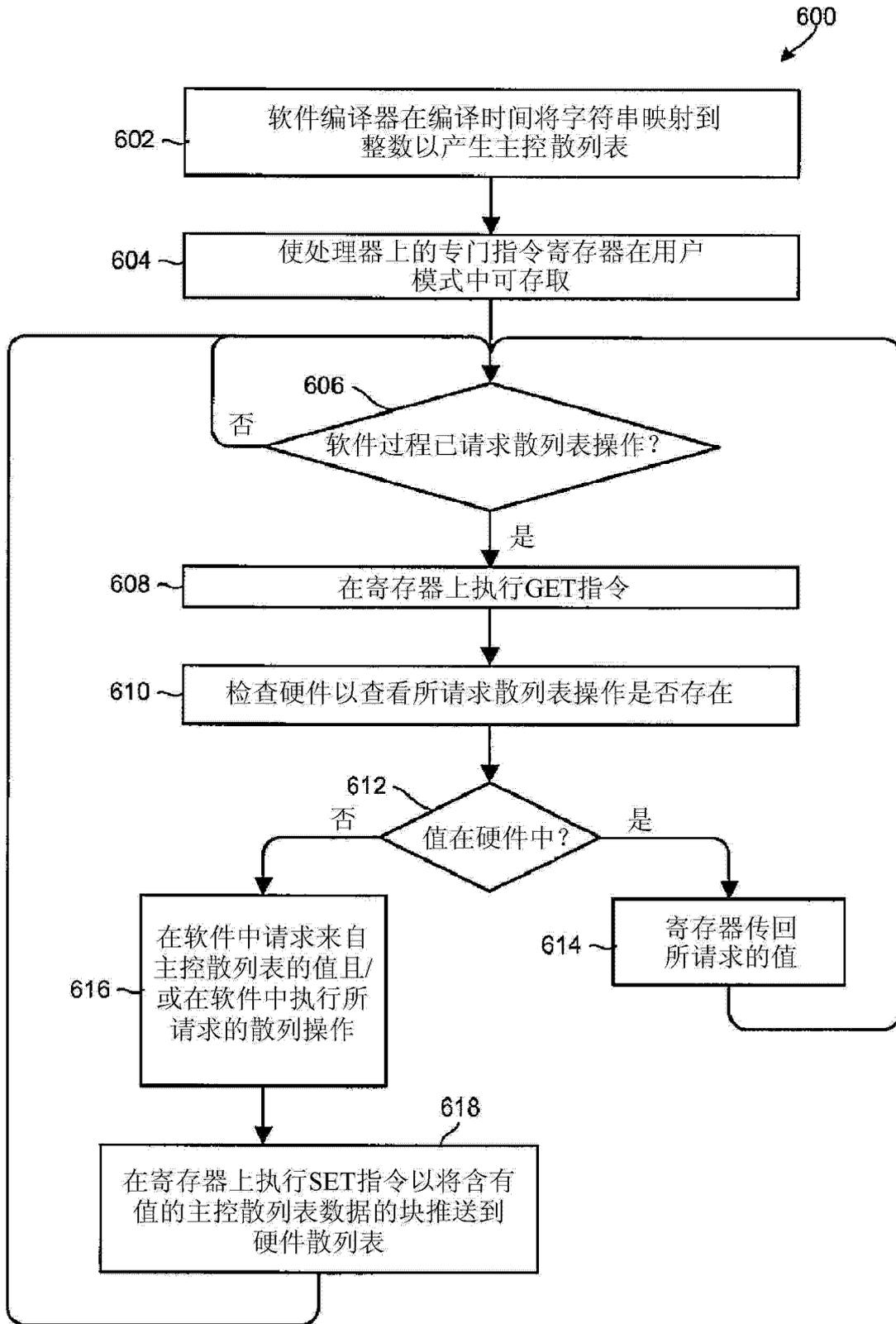


图 6

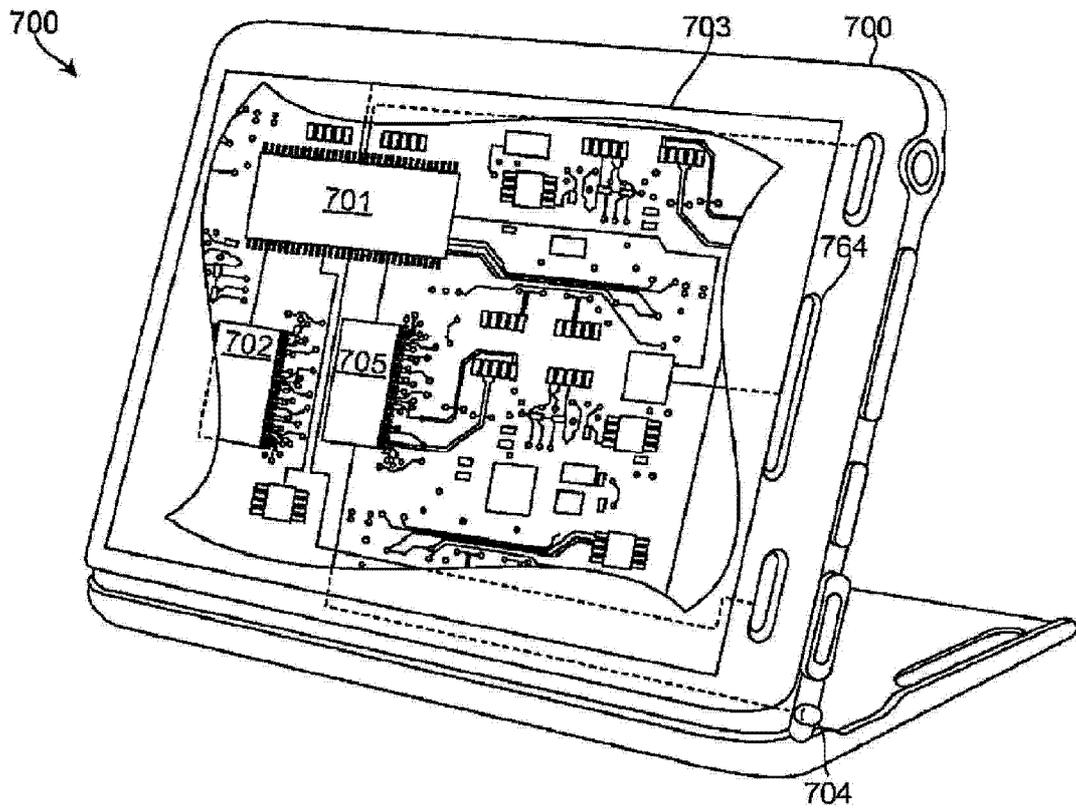


图 7

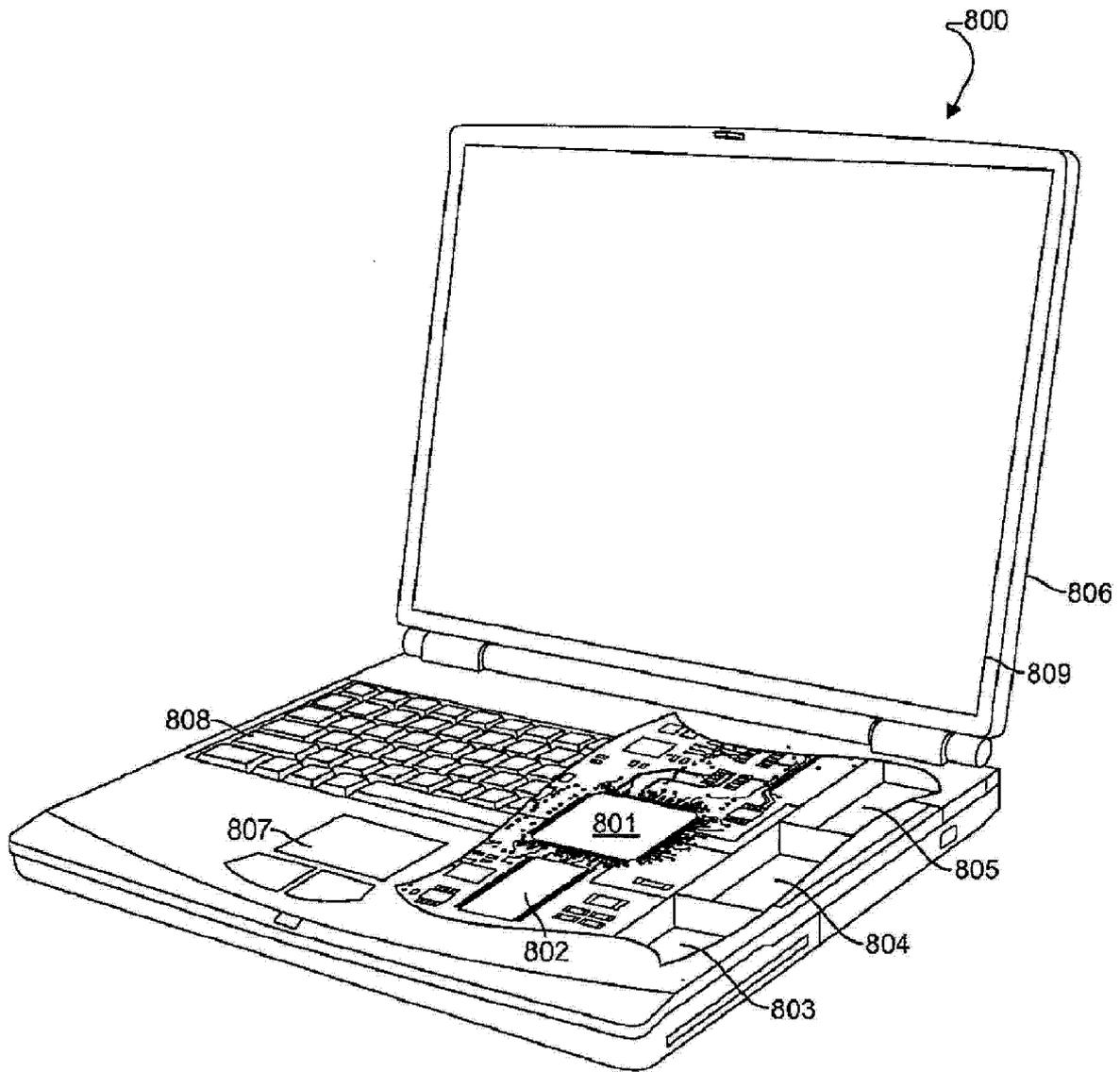


图 8