



(12) 发明专利申请

(10) 申请公布号 CN 105659502 A

(43) 申请公布日 2016. 06. 08

(21) 申请号 201480058159. 7

代理人 张扬 王英

(22) 申请日 2014. 10. 23

(51) Int. Cl.

(30) 优先权数据

H03M 7/40(2006. 01)

14/062, 859 2013. 10. 24 US

H03M 7/30(2006. 01)

(85) PCT国际申请进入国家阶段日

2016. 04. 22

(86) PCT国际申请的申请数据

PCT/US2014/061921 2014. 10. 23

(87) PCT国际申请的公布数据

W02015/061541 EN 2015. 04. 30

(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 D·全 H·J·罗

(74) 专利代理机构 永新专利商标代理有限公司

72002

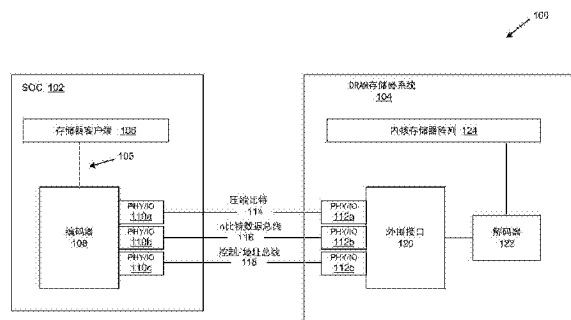
权利要求书3页 说明书11页 附图16页

(54) 发明名称

用于节省存储器系统中的功耗的系统和方法

(57) 摘要

公开了用于节省存储器系统中的功耗的系统和方法。一种这样的系统包括片上系统 (SoC) 和编码器。所述 SoC 包括用于访问耦合到所述 SoC 的动态随机存取存储器 (DRAM) 存储器系统的一个或多个存储器客户端。所述编码器位于所述 SoC 上, 并且被配置为通过根据压缩方案来编码所接收的存储器数据, 来减小从所述存储器客户端接收的存储器数据的数据活动因子, 并且将所编码的存储器数据提供给所述 DRAM 存储器系统。所述 DRAM 存储器系统被配置为根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。



1. 一种用于节省存储器系统中的功耗的方法,所述方法包括:

从位于片上系统(SoC)上的一个或多个存储器客户端接收存储器数据,所述存储器客户端用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统;

通过根据压缩方案来编码所接收的存储器数据,来减小由所接收的存储器数据定义的数据活动因子;

将所编码的存储器数据提供给所述DRAM存储器系统;以及

所述DRAM存储器系统根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。

2. 根据权利要求1所述的方法,其中,所述压缩方案包括熵编码算法。

3. 根据权利要求1所述的方法,其中,所述根据所述压缩方案来编码所接收的存储器数据包括:针对每个由所述DRAM存储器系统定义的最小访问长度(MAL)事务,确定与所接收的存储器数据的预定字节大小相关联的比特模式是否与和所述压缩方案相关联的多个预定码字中的一个预定码字相匹配。

4. 根据权利要求3所述的方法,其中:如果所述比特模式与所述预定码字中的一个预定码字相匹配,那么利用匹配的预定码字来编码所述比特模式。

5. 根据权利要求3所述的方法,其中,所述编码所接收的存储器数据还包括:基于由所述DRAM存储器系统定义的所述MAL事务的大小来补零所编码的存储器数据。

6. 根据权利要求1所述的方法,其中,所述编码所接收的存储器数据还包括:生成压缩比特,所述压缩比特标识针对最小访问长度(MAL)事务的所编码的存储器数据已经被压缩。

7. 根据权利要求6所述的方法,还包括:将所述压缩比特与针对所述MAL事务的所编码的存储器数据级联。

8. 根据权利要求6所述的方法,还包括:独立于所编码的存储器数据来将所述压缩比特提供给所述DRAM存储器系统。

9. 根据权利要求1所述的方法,其中,所述存储器客户端包括中央处理单元(CPU)、图形处理单元(GPU)以及数字信号处理器(DSP)中的一个或多个。

10. 根据权利要求1所述的方法,其中,所述压缩算法包括简化的霍夫曼方案,所述简化的霍夫曼方案包括多个可编程系数。

11. 一种用于节省存储器系统中的功耗的系统,所述系统包括:

用于从位于片上系统(SoC)上的一个或多个存储器客户端接收存储器数据的单元,所述存储器客户端用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统;

用于通过根据压缩方案来编码所接收的存储器数据,来减小由所接收的存储器数据定义的数据活动因子的单元;

用于将所编码的存储器数据提供给所述DRAM存储器系统的单元;以及

用于根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据的单元。

12. 根据权利要求11所述的系统,其中,所述压缩方案包括熵编码算法。

13. 根据权利要求11所述的系统,其中,所述用于根据所述压缩方案来编码所接收的存储器数据的单元包括:用于针对每个由所述DRAM存储器系统定义的最小访问长度(MAL)事务,确定与所接收的存储器数据的预定字节大小相关联的比特模式是否与和所述压缩方案相关联的多个预定码字中的一个预定码字相匹配的单元。

14. 根据权利要求13所述的系统,其中:如果所述比特模式与所述预定码字中的一个预定码字相匹配,那么利用匹配的预定码字来编码所述比特模式。

15. 根据权利要求13所述的系统,其中,所述用于编码所接收的存储器数据的单元还包括:基于由所述DRAM存储器系统定义的所述MAL事务的大小来补零所编码的存储器数据。

16. 根据权利要求11所述的系统,其中,所述用于编码所接收的存储器数据的单元还包括:生成压缩比特,所述压缩比特标识针对最小访问长度(MAL)事务的所编码的存储器数据已经被压缩。

17. 根据权利要求16所述的系统,还包括:用于将所述压缩比特与针对所述MAL事务的所编码的存储器数据级联的单元。

18. 根据权利要求16所述的系统,还包括:用于独立于所编码的存储器数据来将所述压缩比特提供给所述DRAM存储器系统的单元。

19. 根据权利要求11所述的系统,其中,所述存储器客户端包括中央处理单元(CPU)、图形处理单元(GPU)以及数字信号处理器(DSP)中的一个或多个。

20. 根据权利要求11所述的系统,其中,所述压缩算法包括简化的霍夫曼方案,所述简化的霍夫曼方案包括多个可编程系数。

21. 一种包括计算机可用介质的计算机程序产品,所述计算机可用介质具有体现在其中的计算机可读程序代码,所述计算机可读程序代码适于被执行以实现用于节省存储器系统中的功耗的方法,所述方法包括:

从位于片上系统(SoC)上的一个或多个存储器客户端接收存储器数据,所述存储器客户端用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统;

通过根据压缩方案来编码所接收的存储器数据,来减小由所接收的存储器数据定义的数据活动因子;

将所编码的存储器数据提供给所述DRAM存储器系统;以及

所述DRAM存储器系统根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。

22. 根据权利要求21所述的计算机程序产品,其中,所述压缩方案包括熵编码算法。

23. 根据权利要求21所述的计算机程序产品,其中,所述根据所述压缩方案来编码所接收的存储器数据包括:针对每个由所述DRAM存储器系统定义的最小访问长度(MAL)事务,确定与所接收的存储器数据的预定字节大小相关联的比特模式是否与和所述压缩方案相关联的多个预定码字中的一个预定码字相匹配。

24. 根据权利要求23所述的计算机程序产品,其中:如果所述比特模式与所述预定码字中的一个预定码字相匹配,那么利用匹配的预定码字来编码所述比特模式。

25. 根据权利要求23所述的计算机程序产品,其中,所述编码所接收的存储器数据还包括:基于由所述DRAM存储器系统定义的所述MAL事务的大小来补零所编码的存储器数据。

26. 根据权利要求21所述的计算机程序产品,其中,所述编码所接收的存储器数据还包括:生成压缩比特,所述压缩比特标识针对最小访问长度(MAL)事务的所编码的存储器数据已经被压缩。

27. 根据权利要求26所述的计算机程序产品,还包括:将所述压缩比特与针对所述MAL事务的所编码的存储器数据级联。

28. 根据权利要求26所述的计算机程序产品,还包括:独立于所编码的存储器数据来将所述压缩比特提供给所述DRAM存储器系统。

29. 根据权利要求21所述的计算机程序产品,其中,所述存储器客户端包括中央处理单元(CPU)、图形处理单元(GPU)以及数字信号处理器(DSP)中的一个或多个。

30. 根据权利要求21所述的计算机程序产品,其中,所述压缩算法包括简化的霍夫曼方案,所述简化的霍夫曼方案包括多个可编程系数。

31. 一种用于节省存储器系统中的功耗的系统,所述系统包括:

片上系统(SoC),所述SoC包括用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统的一个或多个存储器客户端;

编码器,所述编码器位于所述SoC上,并且被配置为通过根据压缩方案来编码所接收的存储器数据,来减小从所述存储器客户端接收的存储器数据的数据活动因子,并且将所编码的存储器数据提供给所述DRAM存储器系统;以及

所述DRAM存储器系统,所述DRAM存储器系统被配置为根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。

32. 根据权利要求31所述的系统,其中,所述压缩方案包括熵编码算法。

33. 根据权利要求31所述的系统,其中,所述编码器被配置为针对每个由所述DRAM存储器系统定义的最小访问长度(MAL)事务,确定与所接收的存储器数据的预定字节大小相关联的比特模式是否与和所述压缩方案相关联的多个预定码字中的一个预定码字相匹配。

34. 根据权利要求33所述的系统,其中,如果所述比特模式与所述预定码字中的一个预定码字相匹配,那么利用匹配的预定码字来编码所述比特模式。

35. 根据权利要求33所述的系统,其中,所述编码器还被配置为基于由所述DRAM存储器系统定义的所述MAL事务的大小来补零所编码的存储器数据。

36. 根据权利要求31所述的系统,其中,所述编码器还被配置为生成压缩比特,所述压缩比特标识针对最小访问长度(MAL)事务的所编码的存储器数据已经被压缩。

37. 根据权利要求36所述的系统,其中,所述编码器还被配置为将所述压缩比特与针对所述MAL事务的所编码的存储器数据级联。

38. 根据权利要求36所述的系统,其中,所述压缩比特是独立于所编码的存储器数据来被提供给所述DRAM存储器系统的。

39. 根据权利要求31所述的系统,其中,所述存储器客户端包括中央处理单元(CPU)、图形处理单元(GPU)以及数字信号处理器(DSP)中的一个或多个。

40. 根据权利要求31所述的系统,其中,所述压缩算法包括简化的霍夫曼方案,所述简化的霍夫曼方案包括多个可编程系数。

用于节省存储器系统中的功耗的系统和方法

[0001] 相关申请声明

[0002] 本申请涉及于2013年10月24日递交的名为“SYSTEM AND METHOD FOR PROVIDING MULTI-USER POWER SAVING CODEBOOK OPTMIZATION”(高通案号133990U2)的共同未决美国专利申请序列号___。

背景技术

[0003] 动态随机存取存储器(DRAM)被用在各种计算设备(例如,个人计算机、膝上型计算机、笔记本计算机、视频游戏控制台、便携式计算设备、移动电话等)中。DRAM是一种类型的易失性存储器,其将数据的每个比特存储在集成电路内的单独的电容器中。电容器可以被充电或放电。采用这两个状态来表示比特的两个值,常规上被称为0和1。因为电容器漏电,所以信息最终渐弱,除非电容器电荷被周期性地得到补充。因为这个更新要求,所以与SRAM和其它静态存储器相反,DRAM被称为动态存储器。

[0004] DRAM的优势是其结构的简易(每比特仅需要一个晶体管和一个电容器),这允许DRAM达到非常高的密度。然而,随着DRAM密度和速度要求持续增长,存储器功耗正成为重要的问题。

[0005] DRAM内的功率通常被分类为内核存储器阵列功率和非内核功率。内核存储器阵列功率指代用于保留比特单元/阵列中的所有数据并且管理泄露和更新操作的功率。非内核功率指代用于将所有数据传入或传出存储器设备、感测放大器并且管理外围逻辑单元、复用器、内部总线、缓冲器、输入/输出(I/O)驱动器和接收机的功率。降低非内核功率是重要的问题。

[0006] 用于降低非内核功率的现有的解决办法典型地涉及减小操作电压、减小负载电容或每当不要求性能时暂时地降低操作频率。然而,这些解决办法未能解决要求很高的带宽密集型使用情况。其它解决办法尝试减小与存储器系统相关联的数据活动因子。数据活动因子 k 指代在固定时段内存存储器访问系统中的0到1的切换或转变的数量。例如,在下面的单个导线上的8节拍序列0、1、0、1、0、1、0、1中, $k=0.5$ 。已经针对特定类型的数据(诸如使用图像压缩的显示帧缓冲器)提出了对减小数据活动因子的尝试。这典型地在源(即,显示硬件引擎)处执行。然而,这样的解决办法是非常专用的并且受限于这种类型的显示数据,这种类型的显示数据典型地占总DRAM使用的相对小的百分比。因此,在本领域中仍然存在对用于节省DRAM存储器系统中的功耗的改进的系统和方法的需求。

发明内容

[0007] 公开了用于节省存储器系统中的功耗的系统和方法。一个实施例是一种用于节省存储器系统中的功耗的方法。一种这样的方法包括:从位于片上系统(SoC)上的一个或多个存储器客户端接收存储器数据,所述存储器客户端用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统;通过根据压缩方案来编码所接收的存储器数据,来减小由所接收的存储器数据定义的数据活动因子;将所编码的存储器数据提供给所述DRAM存储器系

统;以及所述DRAM存储器系统根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。

[0008] 另一个实施例是一种用于节省存储器系统中的功耗的系统。一种这样的系统包括片上系统(SoC),所述SoC包括用于访问耦合到所述SoC的动态随机存取存储器(DRAM)存储器系统的一个或多个存储器客户端。所述SoC还包括编码器,所述编码器被配置为通过根据压缩方案来编码所接收的存储器数据,来减小从所述存储器客户端接收的存储器数据的数据活动因子,并且将所编码的存储器数据提供给所述DRAM存储器系统。所述DRAM存储器系统被配置为根据所述压缩方案来将所编码的存储器数据解码为所接收的存储器数据。

附图说明

[0009] 在附图中,除非另外指示,否则遍及各个视图,相似的附图标记指代相似的部分。对于具有诸如“102A”或“102B”的字母字符标记的附图标记,字母字符标记可以区分出现在同一附图中的两个相似的部分或元素。当旨在使附图标记涵盖在所有附图中具有相同附图标记的所有部分时,可以省略附图标记的字母字符标记。

[0010] 图1是用于节省耦合到SoC的DRAM存储器系统中的功耗的系统的实施例的框图。

[0011] 图2是示出了将图1的SoC与DRAM存储器系统耦合的数据总线的示例性实施例的图。

[0012] 图3是示出了针对由DRAM存储器系统定义的示例性最小访问长度(MAL)事务的到图1的编码器的未经压缩的数据输入和从图1的编码器的经压缩的数据输出的数据图。

[0013] 图4是示出了在图1的系统中实现的用于节省功耗的方法的实施例的流程图。

[0014] 图5是用于实现用于减小图1的系统的的活动因子的压缩算法的实施例的简化霍夫曼树。

[0015] 图6示出了针对图1的DRAM存储器系统的示例性MAL事务的第一压缩使用情况。

[0016] 图7示出了针对图1的DRAM存储器系统的示例性MAL事务的第二压缩使用情况。

[0017] 图8是示出了图1的SoC中的编码器的实施例的框图。

[0018] 图9是示出了图1的DRAM存储器系统中的解码器的实施例的框图。

[0019] 图10是示出了图8的编码器中的3比特大小输出的示例性值的表。

[0020] 图11是用于跟踪针对图1的系统的压缩统计的表的实施例。

[0021] 图12是包括图1的系统的便携式计算机设备的实施例的框图。

[0022] 图13是用于优化多个用户的编码器压缩性能的系统实施例的框图。

[0023] 图14是示出了由图13的系统中的编码器优化模块生成的服务器数据库的实施例的数据图。

[0024] 图15示出了与计算设备的存储器图像相关联的示例性码本的实施例。

[0025] 图16是示出了图13的系统中的服务器编码器优化模块的实施例的架构、操作和/或功能的流程图。

[0026] 图17是示出了用于生成针对图13的系统中的多个用户的经优化的码本的各种示例性设备度量的表。

具体实施方式

[0027] 本文使用的词语“示例性”意味着“作为示例、实例或说明”。本文中描述为“示例性”的任何方面不必被解释为优选于其它方面或比其它方面有优势。

[0028] 在本描述中,术语“应用”还可以包括具有可执行内容(诸如:对象代码、脚本、字节代码、标记语言文件以及补丁)的文件。另外,本文中所引用的“应用”还可以包括本质上不可执行的文件(诸如可能需要被打开的文档或需要被访问的其它数据文件)。

[0029] 术语“内容”还可以包括具有可执行内容(诸如:对象代码、脚本、字节代码、标记语言文件以及补丁)的文件。另外,本文中所引用的“内容”还可以包括本质上不可执行的文件(诸如可能需要被打开的文档或需要被访问的其它数据文件)。

[0030] 如在本描述中使用的,术语“组件”、“数据库”、“模块”、“系统”等等旨在指代与计算机相关的实体,要么是硬件、固件、硬件和软件的组合、软件,要么是执行中的软件。例如,组件可以是,但不限于是:在处理器上运行的过程、处理器、对象、可执行文件、执行的线程、程序和/或计算机。通过说明的方式,在计算设备上运行的应用和计算设备二者可以是组件。一个或多个组件可以存在于过程和/或执行的线程中,以及组件可以位于一个计算机中和/或分布在两个或更多个计算机之间。另外,这些组件可以从具有存储在其上的各种数据结构的各种计算机可读介质中执行。组件可以诸如根据具有一个或多个数据分组(例如,来自与本地系统、分布式系统中的另一个组件进行交互,和/或跨诸如互联网的网络通过信号的方式与其它系统进行交互的一个组件的数据)的信号,通过本地和/或远程过程的方式进行通信。

[0031] 在本描述中,术语“通信设备”、“无线设备”、“无线电话”、“无线通信设备”和“无线手持机”被可互换地使用。随着第三代(“3G”)无线技术和第四代(“4G”)的出现,更大的带宽可用性已经实现了具有更多种类的无线能力的更加便携的计算设备。因此,便携式计算设备可以包括蜂窝电话、寻呼机、PDA、智能电话、导航设备或具有无线连接或链路的手持计算机。

[0032] 图1示出了用于节省DRAM存储器系统104中的功耗的系统100。系统100可以实现在任何计算设备中,包括个人计算机、工作站、服务器、便携式计算设备(PCD)(诸如蜂窝电话、便携式数字助理(PDA)、便携式游戏控制台、掌上型计算机或平板计算机)。如图1的实施例所示,系统100包括耦合到DRAM存储器系统104的片上系统(SoC)102。SoC 102包括各种片上组件,所述各种片上组件包括向DRAM存储器系统104请求存储器资源的一个或多个存储器客户端106。存储器客户端106可以包括一个或多个处理器单元(例如,中央处理单元(CPU)、图形处理单元(GPU)、数字信号处理器(DSP)、显示处理器等)、视频编码器、或请求到DRAM存储器系统104的读取/写入访问的其它客户端。存储器客户端106经由SoC总线105连接到编码器108。

[0033] 如下文更加详细的描述的,编码器108被配置为通过减小到DRAM存储器系统104的数据输入的数据活动因子 k ,来降低DRAM存储器系统104的功耗。DRAM存储器系统104内的功率可以被分类为内核存储器阵列功率和非内核功率。如本领域已知的,内核存储器阵列功率指代用于保留内核存储器阵列124中的所有数据并且管理泄露和更新操作的功率。非内核功率指代用于将所有数据传入或传出存储器设备、感测放大器并且管理外围逻辑单元、复用器、内部总线、缓冲器、输入/输出(I/O)驱动器和接收机的功率。编码器108通过经由例如基于熵的压缩来减小存储器数据输入的数据活动因子,来降低非内核功率。

[0034] DRAM存储器系统104中的动态或非内核功率可以由公式1来表示:

[0035] 动态功率= kCV^2f *密度,其中:

[0036] k =数据活动因子

[0037] C =负载电容

[0038] V =电压

[0039] f =频率或切换速率

[0040] 密度=以千兆字节为单位的总容量(GB)

[0041] 公式1

[0042] 数据活动因子 k 可以被定义为在固定时段内的0到1的切换或转变的数量。例如,在1比特8节拍序列01010101中, $k=0.5$ 。到DRAM存储器系统104的最小访问被称为一个DRAM最小访问长度(MAL)事务。对于32比特并行LPDDR3DRAM总线, $MAL=32$ 比特*8节拍=256比特=32字节(8节拍、32比特宽度)。MAC事务可以连续地、紧接地发生。

[0043] 因为密度和频率需求在增长,所以减小非内核功率要求:减小负载电容,降低电压,逐节拍地最小化针对每个比特的 k ,或逐MAL地最小化针对每个比特的 k 。用于降低非内核功率的现有的方法通常涉及降低操作电压、减小负载电容,或每当不要求性能时暂时地减小操作频率(其未能解决苛求带宽密集型使用情况)。已经针对特定类型的数据(诸如使用图像压缩的显示帧缓冲器)提出了对减小数据活动因子 k 的尝试。然而,这典型地在源(例如,显示硬件引擎)处执行。然而,这样的解决办法是非常专用的并且受限于这种类型的显示数据,这种类型的显示数据典型地占总DRAM使用的相对小的百分比。

[0044] 图1的系统100可以通过减小所有存储器客户端106的存储器数据输入的数据活动因子 k ,来节省或优化整个DRAM存储器系统104的非内核功耗。SoC 102和DRAM存储器系统104经由一个或多个连接、接口或总线来进行通信。在图1的实施例中,SoC 102包括物理层或输入/输出设备(PHY/IO)110a、110b和110c。DRAM存储器系统104包括PHY/IO 112a、112b和112c。PHY/IO 110a和PHY/IO 112a经由连接114耦合,所述连接114可以包括用于传送元数据压缩比特(被称为“c比特”)以指示数据是否已经被压缩的通道。PHY/IO 110b和PHY/IO 112b经由连接116耦合,所述连接116可以包括 n 比特数据总线。PHY/IO 110c和PHY/IO 112c经由连接118耦合,所述连接118可以包括控制/地址总线。

[0045] 在操作中,来自SoC 102内的存储器客户端106的存储器数据经过编码器108。编码器108可以经由例如用于压缩和补零数据的简化的霍夫曼(Huffman)方案来压缩存储器数据,所述存储器数据随后被经由连接114和116提供给DRAM存储器系统104。DRAM存储器系统104将数据接收到PHY/IO设备112a、112b和/或112c。外围接口120将所压缩的数据提供给解码器122,所述解码器122被配置为将数据反变换回原始的未经压缩的形式,并且随后被存储到内核存储器阵列124。应当认识到的是,DRAM存储器系统104可以包括任意数量的为任何期望类型、大小和配置的存储器的DRAM存储器设备。

[0046] 图2示出了在SoC 102和DRAM存储器系统104之间的示例性32比特并行LPDDR3DRAM总线116。在该实施例中,用于示例性的目的,MAL事务204包括针对时钟202的8节拍(即, $t=0,1,2,3,4,5,6,7$)的32比特。每个 $MAL=32$ 比特*8节拍=256比特=32字节。编码器108通过压缩每个MAL事务204来减小数据活动因子 k 。图3示出了MAL事务204的示例性编码示例。未经压缩的数据302可以经由例如基于熵的压缩来处理以产生经压缩的数据304。未经压缩的

数据302可以包括例如32字节的原始未经压缩的数据。在编码器108中体现的压缩算法可以将未经压缩的数据302压缩为例如16字节,跟在其后的是由被灰化的节拍(即, $t=4,5,6,7$)表示的补零,由此减小了与MAL事务204相关联的数据活动因子 k 。

[0047] 图4示出了由图1的系统100实现的用于减小DRAM存储器系统104的非内核功率的方法400。在框402处,编码器108从位于SoC 102上的、用于访问DRAM存储器系统104的一个或多个存储器客户端106接收存储器数据。在框404处,编码器108通过根据压缩方案来编码所接收的存储器数据,来减小由所接收的存储器数据定义的数据活动因子 k 。在一个实施例中,以逐MAL地为基础减小数据活动因子 k 。应当认识到的是,可以实现压缩方案的各种实施例。在一个实施例中,压缩方案包括经由例如具有补零的简化的霍夫曼方案的基于熵的压缩。在框406处,所编码或压缩的存储器数据被提供给DRAM存储器系统104。如下文更加详细的描述的,编码器108可以包括用于评估用于每个MAL的压缩算法的有效性的逻辑单元。以这种方式,编码器108可以生成压缩或C比特来标识数据是否已经被压缩。在框408处,DRAM存储器系统104中的解码器122可以根据压缩方案,来将所编码的存储器数据解码为原始的所接收的存储器数据。以这种方式,DRAM存储器系统104的非内核功率可以被有选择地降低,以适应较低功率使用情况。

[0048] 图5示出了可以由编码器108实现的基于熵的编码算法的实施例。霍夫曼编码方案可以包括用于编码源符号的码表。码表可以包括基于发生的经估计的概率的预定数量的源符号。简化的霍夫曼树500可以体现要被压缩的最频繁的符号或“模式”。在一个实施例中,算法以每字节为基础来操作。如果与存储器数据相关联的源符号或模式(例如,MAL节拍)与在霍夫曼树500的左半部上的“树叶”中的任何“树叶”相匹配,那么发生字节压缩。应当认识到的是,在附图中,前缀“0x”表示其后为十六进制(hex)数位,而后缀“b”表示前面为二进制数位(比特)。框504表示将模式“00”十六进制与码字(CW=01b)匹配。框506表示将模式“FF”十六进制与码字(CW=001b)匹配。框508表示将模式“0F”十六进制与码字(CW=0001b)匹配。框510表示模式将“F0”十六进制与码字(CW=00001b)匹配。框512表示将模式“55”十六进制与码字(CW=000001b)匹配。框514表示将模式“AA”十六进制与码字(CW=000000b)匹配。应当认识到的是,可以对模式编程。如在图5中进一步示出的,如果不存在匹配,那么霍夫曼树的右半部可能遭受每字节一额外比特惩罚。例如,框502示出模式“XX”可能遭受一比特惩罚并且是利用码字(CW=1b+0x“XX”)来编码的,这产生了9比特的码字长度。

[0049] 图6示出了针对使用霍夫曼树500来压缩的MAL事务的第一压缩使用情况的示例。未经压缩的MAL 602包括32字节的原始数据。8个节拍中的每个节拍包括源模式“00”十六进制,其可以是利用码字(CW=01b)来编码的。在该“最好情况”示例中,作为结果的经压缩的MAL 604包括经压缩的8个字节,其后跟有补零。在图6的示例中,未经压缩的MAL 602中的行中的每行表示源模式“00”。未经压缩的MAL 602中的每行是利用对应的码字(CW=01b)来编码的。经压缩的MAL 604表示对未经压缩的MAL 602中的每行进行编码的结果。

[0050] 图7示出了“最坏情况”示例,其中未经压缩的MAL 702中的每个节拍包括源模式“XX”十六进制,其不被压缩并且遭受每字节一额外比特惩罚。作为结果的经压缩的MAL 704包括36个字节。在图7的示例中,未经压缩的MAL 702中的行中的每行表示源模式“XX”。经压缩的MAL 704表示对未经压缩的MAL 702中的每行进行编码的结果。换句话说,每个源模式“XX”利用码字(CW=1b+0xXX)来编码。在该示例中,在压缩产生较大大小的情况下,编码器

108可以发送未经压缩的数据702,而不发送经压缩的数据704。在这点上,应当认识到的是,编码器108可以生成额外的压缩或C比特,以定义数据是否被压缩。

[0051] 在某些实施例中,C比特可以被单独地发送(例如,经由图1的接口114)并且被存储在DRAM存储器系统104中的单独的存储器设备中。在其它实施例中,C比特可以与在数据总线(例如,图1的接口116)上发送的数据级联,并且被存储在相同的DRAM芯片中。应当进一步认识到的是,C比特可以仅用于接口,而不需要将其存储在DRAM存储器系统104中。如果没有将C比特存储在DRAM存储器中,那么在一实施例中,解码器122可以并入到每个存储器中,如图1所示。如果将C比特存储在DRAM存储器中,那么应当认识到的是,在一实施例中,额外的DRAM空间可以用于存储例如针对数据的每个32字节的1个C比特,并且解码器122可以位于SoC 102中,而不是位于DRAM存储器系统104中。

[0052] 可以利用用于使用例如在系统100内的客户端或外部组件(诸如,例如基于云的服务器)上运行的优化程序来分析压缩系数集合(即,C比特)统计的有效性的逻辑单元,来增强系统100。在一实施例中,编码器108可以包括保持对压缩统计的跟踪并且跨越大量的最终用户来进行改进的计数器。编码器108可以被配置具有关闭针对特定的客户端106的压缩的能力。

[0053] 在一实施例中,DRAM存储器系统104可以被SoC 102上的所有存储器客户端106使用。以这种方式,编码器108在来自所有存储器客户端106的所有业务的路径中。可能存在可能不期望编码来自某个客户端106的数据的实例。例如,如果显示处理器已经压缩了DRAM数据,那么使编码器108重新尝试压缩会是对功率的浪费。因此,编码器108将具有单独的启用比特,并且还将收集针对每个客户端106的C比特统计。在每个DRAM事务期间的每个存储器客户端106可以包括唯一地标识该客户端的主ID(MID)。对于每个存储器客户端106,当其被启用用于压缩时,编码器108可以尝试压缩,并且其可以对事务的总数和未经压缩的事务的数量进行计数。这些计数器/统计对于CPU可以是可用的。默认可以是总是启用针对所有存储器客户端106的压缩。

[0054] 为了禁用压缩,CPU可以清除针对特定的存储器客户端106的启用比特,并且从那时起,到DRAM存储器系统104的任何写入可以绕过编码器108,但是C比特可以仍然作为0被发送,这意味着数据是未经压缩的。从DRAM存储器系统104的任何读取可以包含经压缩的或未经压缩的数据,并且C比特可以正确地指示是否需要解压缩。例如,甚至在CPU已经清除了针对特定的存储器客户端106的压缩启用比特之后,仍然可以发生对所读取的数据的解压缩。

[0055] 图11示出了可以由CPU访问的示例性的表1100。表1100包括客户端名称字段1102、主ID(MID)字段1104、压缩启用比特字段1106、事务总数字段1108以及未经压缩的事务总数字段1109。每个存储器客户端106具有唯一的MID。CPU可以启用或禁用针对每个客户端的压缩。当被启用时,编码器108可以保持针对每个客户端的压缩统计的经更新的计数,所述计数可以基于针对每个相应的客户端的业务的可压缩性来独立地启用或禁用压缩。例如,在一实施例中,如果特定的客户端具有超出可编程门限的足够的不可压缩的业务(C比特=0),那么可以禁用针对该客户端的压缩。

[0056] 图8和9分别示出了编码器108和解码器122的实施例。编码器108可以包括可编程霍夫曼系数表804、级联/缓冲器810、补零组件814、计数器818以及C比特生成器820。编码器

108在连接802上接收未经压缩的数据输入。在该示例中,未经压缩的数据包括32字节(8比特),如上所述。表804包括可以用于实现例如霍夫曼树500(图5)的可编程编码器系数。编码器系数可以是例如在重启期间从CPU加载的。CPU可以执行存在于ROM或辅加载器中的未经压缩的代码。霍夫曼输出(9比特)在连接806上被提供给级联/缓冲器810,所述级联/缓冲器810将经级联的输出(8比特)经由连接812提供给补零组件814。补零组件814将经压缩的输出(8比特)提供给连接816再到解码器122(图9)。

[0057] 大小(3比特)被经由连接808提供给计数器818。图10是示出了3比特表示(值0-7)和其相应的定义的表1000。C比特生成器820可以被配置为确定何时达到预定的字节大小。C比特生成器820生成C比特并且经由连接822提供C比特,以标识连接816上的数据输入是否已经被压缩。如上所述,如果压缩产生了较大大小,那么可以将C比特设置为C=0,这指示原始数据输入是输出,这是因为原始数据输入比经压缩的数据小(例如,经压缩的大小>未经压缩的大小)。

[0058] 参照图9,经压缩的数据和C比特可以分别经由连接816和822被缓冲器&左偏移组件902接收。经偏移的输出(8比特)可以经由连接904被提供给可编程反霍夫曼系数表905,所述可编程反霍夫曼系数表905包括由CPU加载的反系数。经解压缩的数据输出可以被经由连接908提供给内核存储器阵列124。

[0059] 如上所述,系统100可以并入到任何期望的计算系统中。图12示出了系统100并入到示例性的便携式计算设备(PCD)1200中。将易于认识到的是,系统100的某些组件(例如,编码器108)被包括在SoC 322(图12)上,而其它组件(例如,DRAM存储器系统104)是耦合到SoC 322的外部组件。SoC 322可以包括多内核CPU 402A。多内核CPU 1202可以包括第零内核410、第一内核412和第N内核414。内核中的一个内核可以包括例如图形处理单元(GPU),以及其它内核中的一个或多个内核包括CPU。

[0060] 显示器控制器328和触摸屏控制器330可以耦合到CPU 1202。继而,在片上系统322外部的触摸屏显示器108可以耦合到显示器控制器1206和触摸屏控制器330。

[0061] 图12还示出了视频编码器334(例如,逐行倒相(PAL)编码器、顺序传送彩色与存储制(SECAM)编码器或国家电视系统委员会(NTSC)编码器)耦合到多内核CPU 1202。此外,视频放大器336耦合到视频编码器334和触摸屏显示器1206。此外,视频端口338耦合到视频放大器336。如图12所示,通用串行总线(USB)控制器340耦合到多内核CPU 1202。此外,USB端口342耦合到USB控制器340。存储器1204和用户身份模块(SIM)卡346也可以耦合到多内核CPU 1202。存储器1204可以位于SoC 322上或耦合到SoC 322(如图1所示)。存储器1204可以包括DRAM存储器系统104(图1),如上所述。

[0062] 进一步地,如图12所示,数字相机348可以耦合到多内核CPU 1202。在一示例性的方面中,数字相机348是电荷耦合器件(CCD)相机或互补金属氧化物半导体(CMOS)相机。

[0063] 如图12进一步所示的,立体声音频编解码器(CODEC)350可以耦合到多内核CPU 1202。此外,音频放大器352可以耦合到立体声音频CODEC 350。在一个示例性的方面中,第一立体声扬声器354和第二立体声扬声器356耦合到音频放大器352。图12示出麦克风放大器358也可以耦合到立体声音频CODEC 350。另外,麦克风360可以耦合到麦克风放大器358。在一特定的方面中,调频(FM)无线电调谐器362可以耦合到立体声音频CODEC 350。此外,FM天线364耦合到FM无线电调谐器362。此外,立体声耳机366可以耦合到立体声音频CODEC

350。

[0064] 图12还示出射频(RF)收发机368可以耦合到多内核CPU 402A。RF开关370可以耦合到RF收发机368和RF天线372。如图12所示,小键盘204可以耦合到多内核CPU 1202。此外,具有麦克风的单声道耳麦376可以耦合到多内核CPU 1202。此外,振动器设备378可以耦合到多内核CPU 1202。

[0065] 图12还示出电源380可以耦合到片上系统322。在一特定的方面中,电源380是向PCD 1200的需要电力的各个组件提供电力的直流(DC)电源。此外,在一特定的方面中,电源是可再充电的DC电池,或者是从连接到交流(AC)电源的AC到DC变压器得到的DC电源。

[0066] 图12还示出PCD 1200还可以包括网卡388,其可以用于接入数据网,例如,局域网、个域网或任何其它网络。网卡388可以是蓝牙网卡、WiFi网卡、个域网(PAN)卡、个域网超低功率技术(PeANUT)网卡、电视/电缆/卫星调谐器、或本领域公知的任何其它网卡。此外,网卡388可以并入到芯片中,即,网卡388可以是芯片中的完全解决方案,以及可以不是单独的网卡388。

[0067] 如图12中描绘的,触摸屏显示器1206、视频端口338、USB端口342、相机348、第一立体声扬声器354、第二立体声扬声器356、麦克风360、FM天线364、立体声耳机366、RF开关370、RF天线372、小键盘374、单声道耳麦376、振动器378以及电源380是在片上系统322的外部的。

[0068] 应当认识到的是,本文描述的方法步骤中的一个或多个方法步骤可以作为计算机程序指令(诸如上述模块)存储在存储器中。这些指令可以由任何适当的处理器结合或与相应的模块一起来执行,以执行本文描述的方法。

[0069] 如上所述,可以由基于云的服务器来优化系统100所实现的压缩方案。图13示出了用于优化在并入到多个计算设备1302中的系统100中实现的压缩算法(例如,码表、压缩系数等)的计算机系统1300的实施例。计算机系统1300包括服务器1306,所述服务器1306经由通信网络1308与多个计算设备1302相通信。每个计算设备1302可以由相应的用户1304来操作。通信网络1308可以支持经由任何适当的协议(包括例如互联网、公共交换电话网(PSTN)、广域网、局域网、无线接入点或任何其它适当的通信基础设施)的有线和/或无线通信。

[0070] 计算设备1302可以包括个人计算机、膝上型计算机、笔记本计算机、视频游戏控制台、便携式计算设备、移动电话等。如图13所示,计算设备1302包括用于通过根据压缩方法来编码存储器数据,来节省存储器系统中的功耗的系统100,如上所述。服务器1306经由通信网络1308与计算设备1302中的每个计算设备进行通信。

[0071] 通常,计算机系统1300包括编码器优化模块,所述编码器优化模块包括用于生成和优化被提供给计算设备1302和由相应的编码器108实现的码本的逻辑和/或功能。应当认识到的是,编码器优化模块的某些方面可以位于计算设备1302处,而其它方面可以位于服务器1306处。客户端侧功能可以由客户端编码器优化模块1310来提供,并且服务器侧功能可以由服务器编码器优化模块1314来提供。在一实施例中,客户端编码器优化模块1310可以包括移动应用以及用户接口特征和控制,所述移动应用提供与服务器1314的数据通信和同步。例如,用户1304可以选择性地启用和禁用码本优化。如下文更加详细的描述的,客户端编码器优化模块1310可以控制码本优化数据到服务器1306(例如,压缩统计和各种设备

和/或用户度量)的传输。通常,服务器编码器优化模块1306包括用于以下操作的逻辑和/或功能:从计算设备1302接收码本优化数据,生成码本并且将码本提供给每个计算设备1302,以及经由数据库1316来优化跨越多个用户1304的网络的码本。

[0072] 图14示出了服务器数据库1316的实施例。服务器数据库1316存储针对计算机系统1302中的每个用户1304的、具有与用户1304和/或相应的计算设备1302相关联的以下信息中的一个或多个信息各种类型的数据:设备存储器图像1404,被提供给计算设备1302的码本1406,以及从计算设备1302接收的码本压缩统计1407和设备/用户度量1408。数据库1316中的每行对应于与计算机系统1300中的不同的用户1304相关联的数据。第一行对应于用户1304a。第二行对应于用户1304b。第三行对应于用户1304c。最后一行对应于用户1304n。应当认识到的是,可以存储任意数量的行以容纳任意数量的用户。

[0073] 图16示出了服务器编码器优化模块1306的实施例的架构、操作和/或功能。在框1602处,可以生成针对计算机系统1300中的每个用户1304的唯一的码本。每个码本与计算设备1302中的一个计算设备相关联,如上所述,并且每个码本被配置用于根据压缩方案来编码相应的计算设备1302中的存储器数据。压缩方案可以包括基于熵的编码算法,诸如在图5中示出的霍夫曼编码方案。如图15所示,码本1406包括用于标识要被压缩的最频繁的符号或“模式”的码表,其中为每个模式分配相应的码字1504。

[0074] 可以通过构建计算设备1302的虚拟存储器图像1404来生成针对计算设备1302的初始码本1406。服务器1306可以接收针对各种软件组件(例如,应用、应用框架、服务/运行时间环境、库、内核、操作系统等)的各种类型的信息(例如图17的信息1700)。服务器1306可以解压缩应用和其它预压缩的结构,并且构建虚拟存储器图像1404。

[0075] 应当认识到的是,可以以各种方式来生成码本1406。在一个实施例中,服务器1306采用分阶段的码本生成过程。第一阶段涉及基于每个软件组件内的模式的静态分布来生成一阶静态码本。服务器1306可以搜索虚拟存储器图像1404中的每个分量,以得到最为重复的代码模式1502,并且为这些代码模式分配最短的码字1504。也可以为频繁地运行过程分配最短的码字1504。第二阶段可以涉及动态码本生成和验证。可以在服务器1306上运行的虚拟设备上加载和脚本编写/执行虚拟存储器图像1404。可以将存储器事务记入日志并且记录读取/写入业务。可以基于动态分布模式而不是静态分布模式来执行类似的模式搜索。

[0076] 再次参照图16,在框1604处,服务器1306将唯一的码本1406经由通信网络1308提供给相应的计算设备1302。计算设备1302可以接收码本1406并且开始使用码本1406来压缩存储器数据,如上所述。在框1606处,服务器1306可以从计算设备1302接收压缩统计和/或设备度量。压缩统计可以包括例如C比特统计,如图11所示。

[0077] 图17示出了信息1700的各种示例,诸如,例如设备度量1702以及可以用于优化码本1406的值1704、1706和1708。可以将压缩统计和设备度量存储在数据库1316中。第一设备度量1702可以包括过程标识符(Process_IDx),所示过程标识符标识了请求存储器资源的特定的过程或任务,并且所述过程标识符可以包括时间戳的值,过程或任务运行的平均时间(%avg_time_running),以及与过程或任务相关联的版本信息。第二设备度量1702可以包括硬件标识符(Phone_Hardware_ID),所述硬件标识符可以包括用于标识硬件型号(Hardware_model)和任何电话修正(Phone_revision)的值。第三设备度量1702可以包括具有用于跟踪时间戳和平均CPU利用的值的CPU利用。第四设备度量1702可以包括针对根据主

ID 1104(图11)识别的特定的客户端的压缩统计。第五设备度量1702可以包括软件标识符(Phone_Software_ID),所述软件标识符可以包括用于标识版本信息的值。

[0078] 应当认识到的是,多个过程可以同时地运行,并且可以接收与计算设备1302相关联的大量额外的度量。在一实施例中,诸如电话硬件ID和电话软件ID的度量可以用于单独地交叉引用,并且从数据库1316在本地获得默认出厂软件,以创建默认虚拟存储器图像1404,以及诸如过程ID和版本的度量可以用于单独地交叉引用,并且从数据库1316在本地获得用户1304已安装的任何额外的软件,并且随后修改出厂虚拟存储器图像1404,以创建特定于用户的虚拟存储器图像1404。在一实施例中,这可以利用极大地减小的通信网络1308带宽来完成,这是因为用户1304的计算设备1302上的实际的图像1404不是直接被发送到服务器1306。可以利用新的软件组件来周期地更新本地数据库1316。

[0079] 在框1608处,服务器1306可以处理来自计算机系统1304中的用户1304中的每个用户的压缩统计和/或设备度量,并且生成针对计算设备1302中的一个或多个计算设备的经优化的码本1406。在一实施例中,服务器1306可以跨越所有具有类似的设备度量的用户1304来寻找,并且希望得到具有最大的成功压缩百分比的C比特统计,这可以转化为改善的功率节省。在框1610处,可以将经优化的码本1406提供给计算设备1302中的一个或多个计算设备。

[0080] 在本说明书中描述的过程或处理流程中的某些步骤自然地先于其它步骤,以使本发明如所描述的运作。但是,本发明不限于所描述的步骤的次序,如果这样的次序或顺序不改变本发明的功能的话。即,认识到的是,在不脱离本发明的范围和精神的情况下,某些步骤可以在其它步骤之前、之后或与其它步骤并行地(大体同时地)执行。在某些实例中,可以在不脱离本发明的情况下省略或不执行某些步骤。此外,诸如“其后”、“随后”、“接下来”等的词语不旨在限制步骤的次序。这些词语仅用于引导读者浏览示例性的方法的描述。

[0081] 另外,编程领域的普通技术人员能够基于例如在本说明书中的流程图和相关联的描述,在没有困难的情况下编写计算机代码或识别适当的硬件和/或电路,以实现所公开的发明。

[0082] 因此,对程序代码指令的特定集合或详细的硬件设备的公开不被认为是获得对如何实现以及使用本发明的足够的理解所必须的。在上文描述中并且结合附图更加详细地解释了所要求保护的计算机实现的过程的发明性功能,所述附图可以说明各个过程流。

[0083] 在一个或多个示例性的方面中,所描述的功能可以在硬件、软件、固件或其任意组合中实现。如果在软件中实现,则所述功能可以作为一个或多个指令或代码存储在计算机可读介质中或者通过其进行传输。计算机可读介质包括计算机存储介质和通信介质二者,所述通信介质包括促进计算机程序从一个地方传送到另一个地方的任何介质。存储介质可以是可由计算机存取的任何可用的介质。通过举例而非限制性的方式,这样的计算机可读介质可以包括RAM、ROM、EEPROM、NAND闪存、NOR闪存、M-RAM、P-RAM、R-RAM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁存储设备、或者可以用于以指令或数据结构的形式携带或存储期望的程序代码以及可以由计算机来存取的任何其它介质。

[0084] 此外,任何连接被适当地称为计算机可读介质。例如,如果使用同轴电缆、光纤光缆、双绞线、数字用户线(“DSL”)或无线技术(例如红外、无线和微波)从网站、服务器或其它远程源发送软件,那么同轴电缆、光纤光缆、双绞线、DSL或无线技术(例如红外、无线和微

波)包括在介质的定义中。

[0085] 如本文所使用的,磁盘和光盘包括压缩光盘(“CD”)、激光光盘、光盘、数字多功能光盘(“DVD”)、软盘和蓝光光盘,其中磁盘通常磁性地复制数据,而光盘则利用激光来光学地复制数据。上述的组合也应当包括在计算机可读介质的范围内。

[0086] 在不脱离本发明的精神和范围的情况下,替代的实施例对于与本发明所属领域的普通技术人员将是显而易见的。因此,尽管详细地说明和描述了所选择的方面,但是将理解的是,如下面的权利要求书所限定的,可以在不脱离本发明的精神和范围的情况下在其中做出各种替代和改变。

100

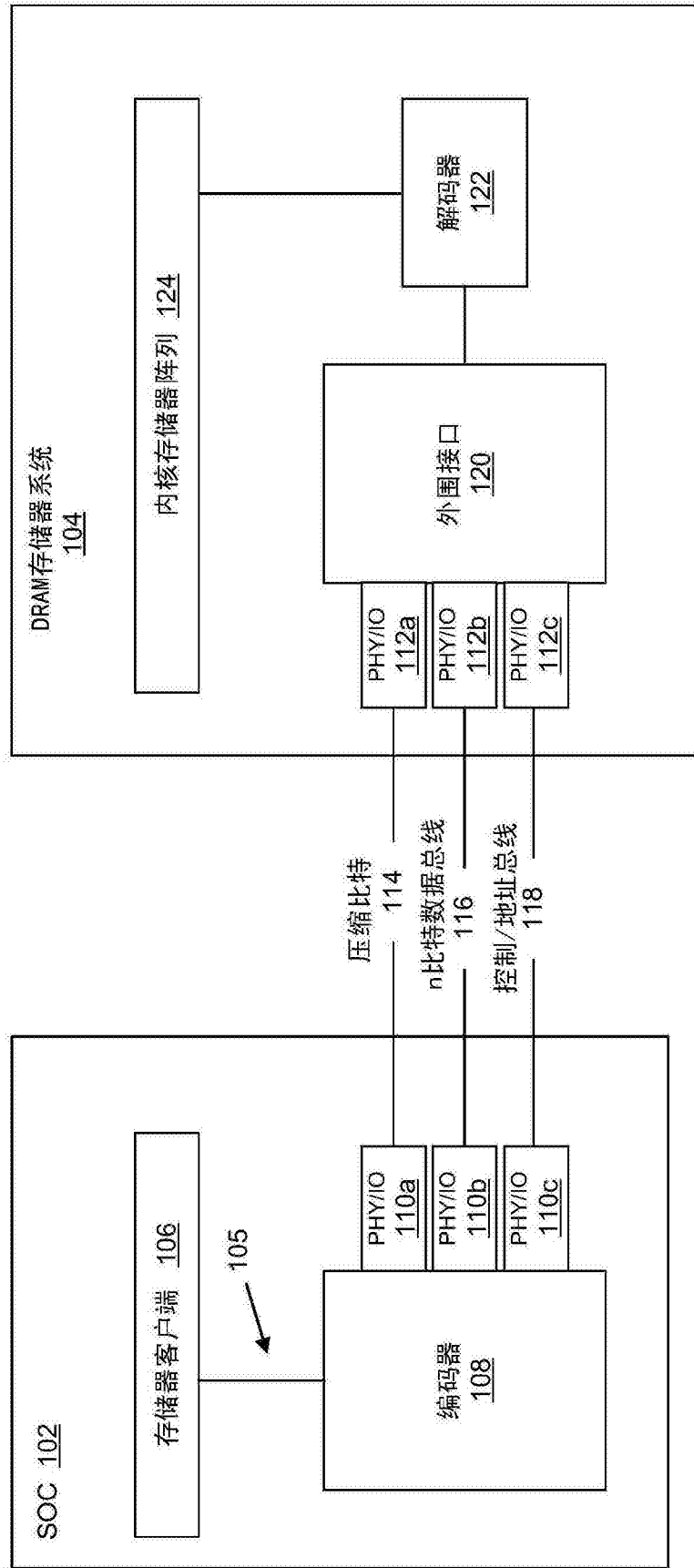


图1

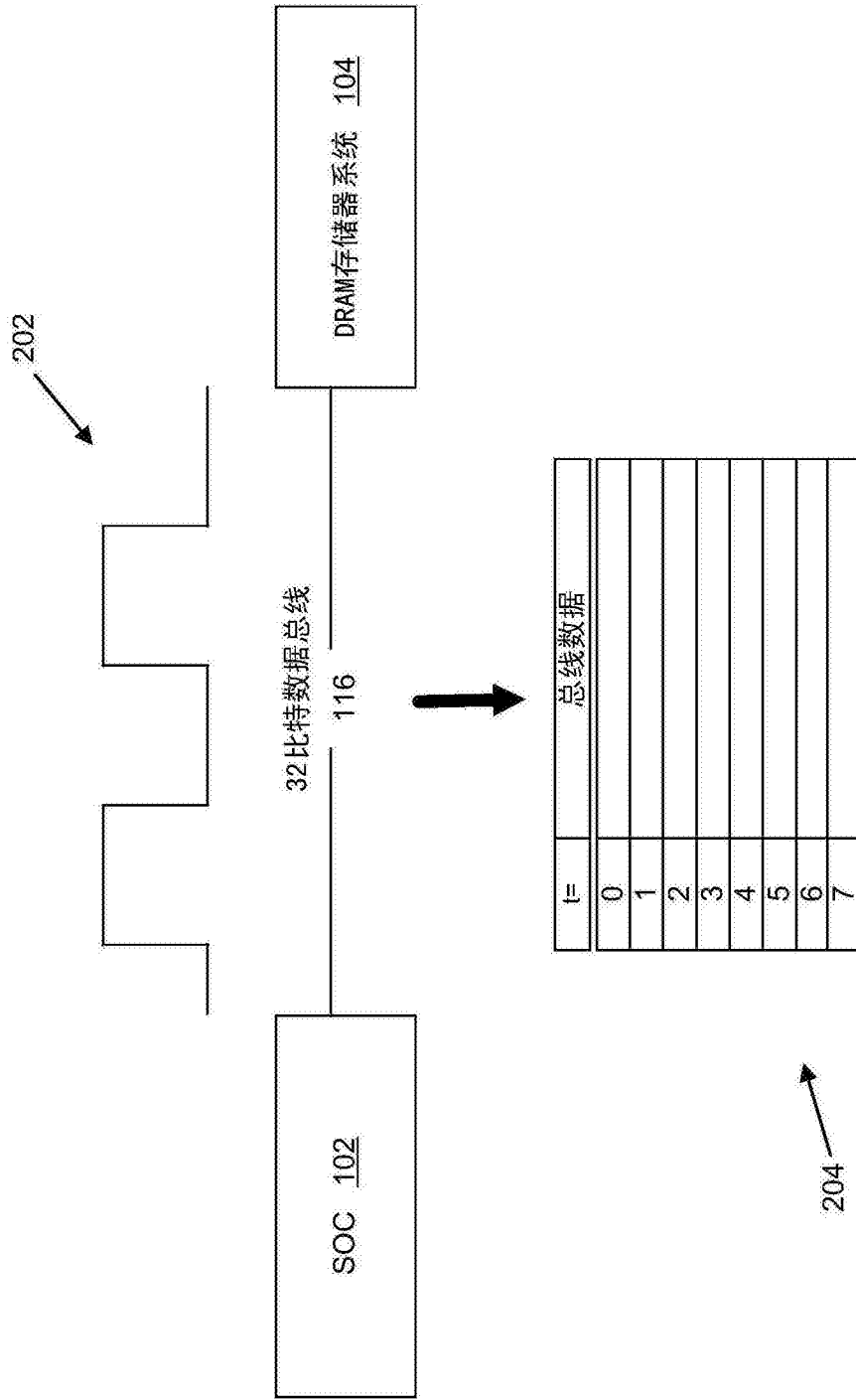


图2

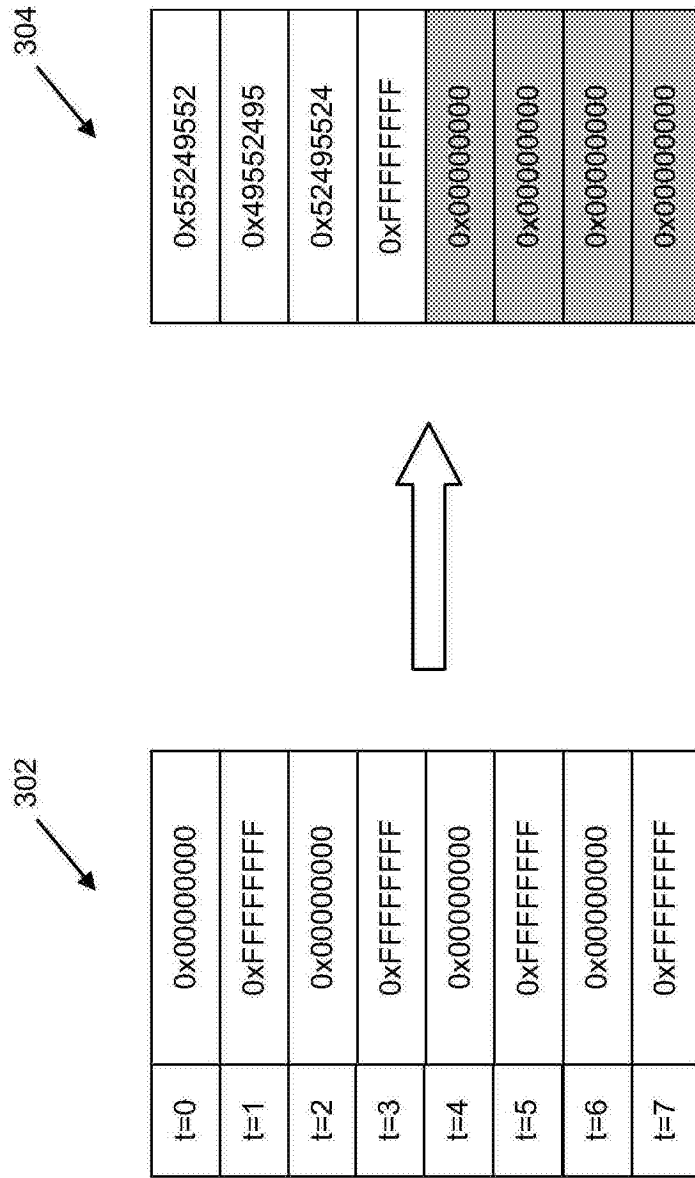


图3

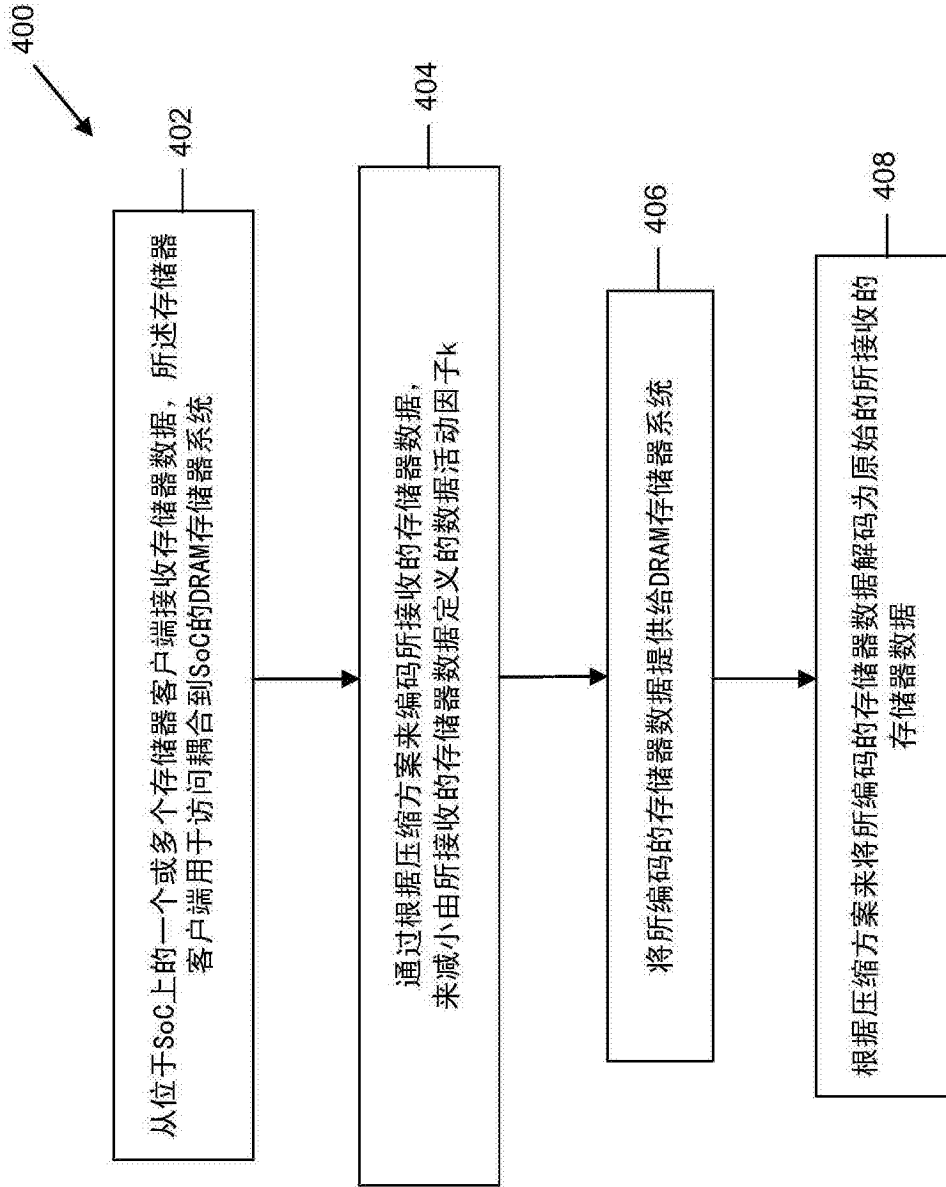


图4

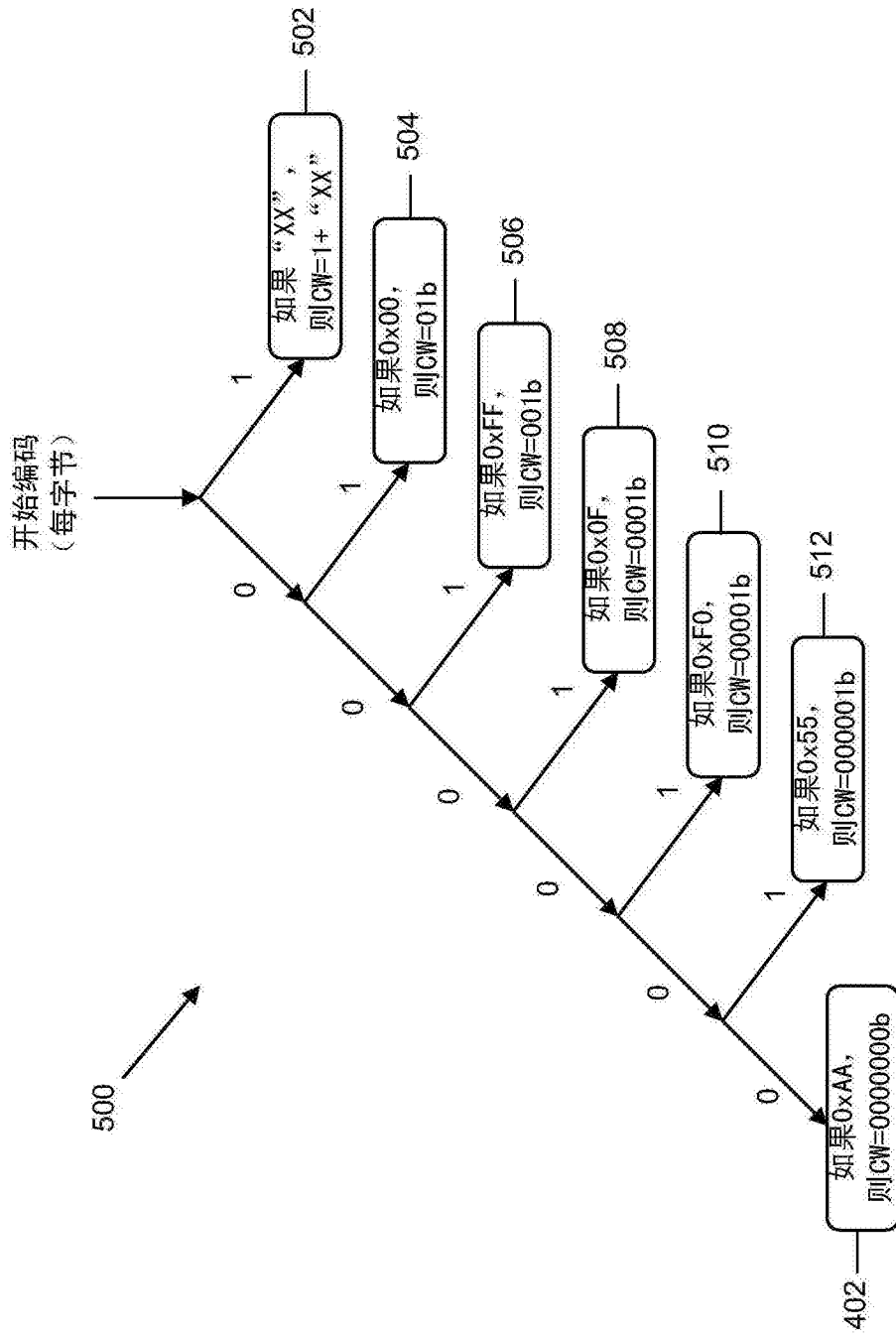


图5

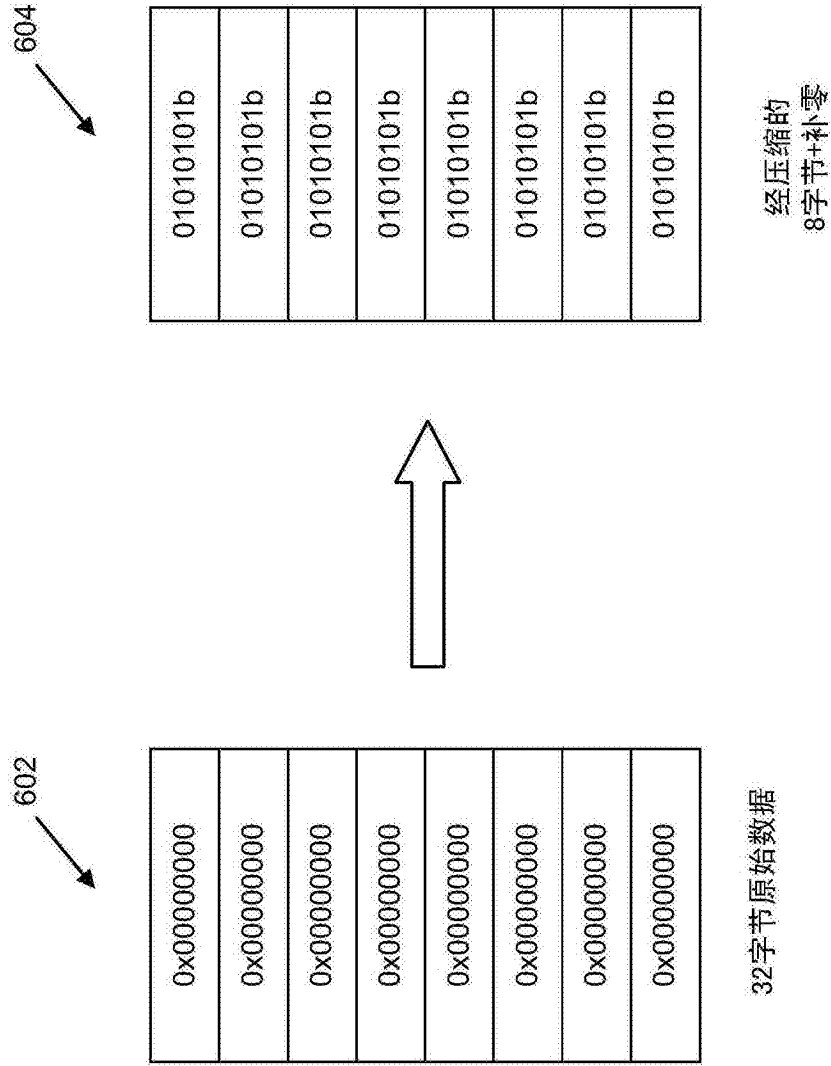


图6

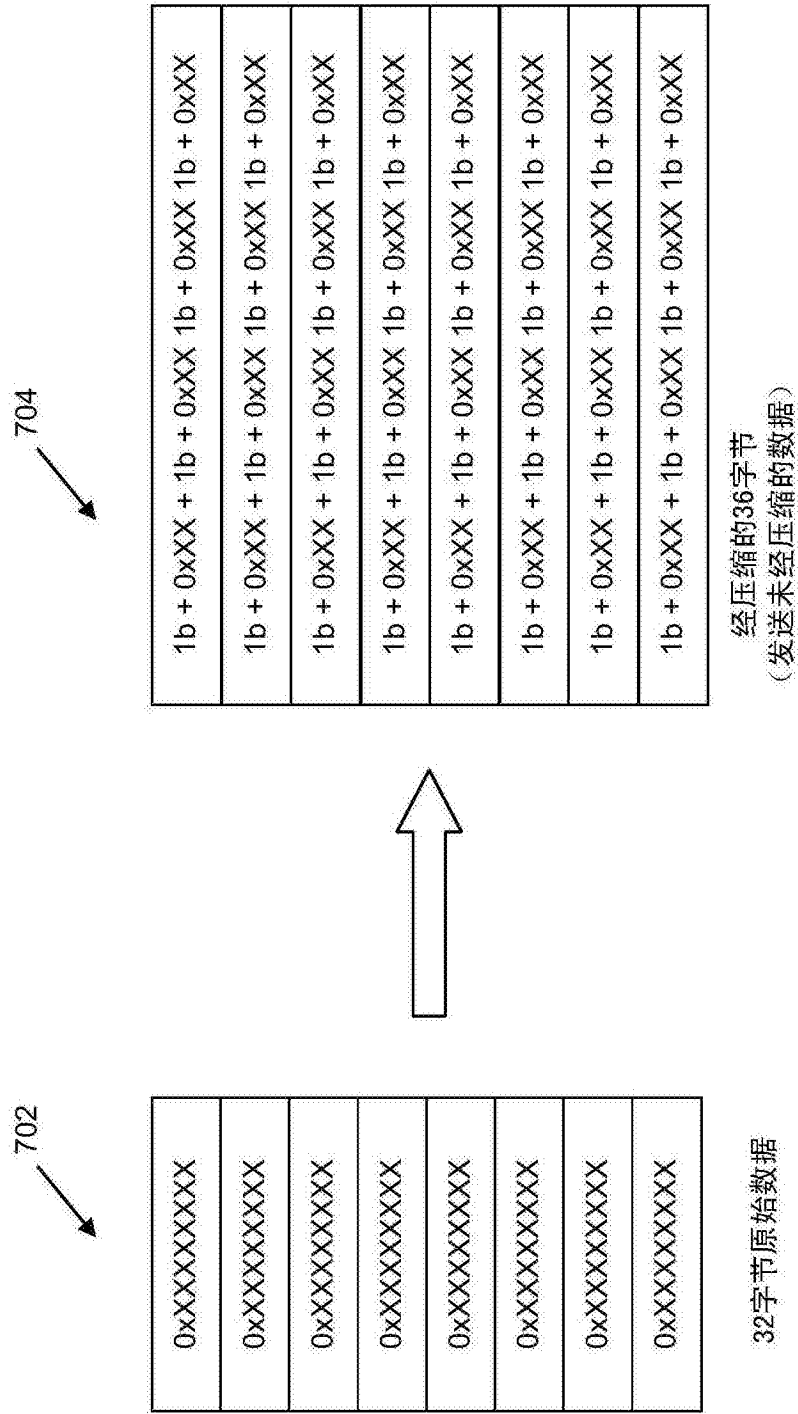


图7

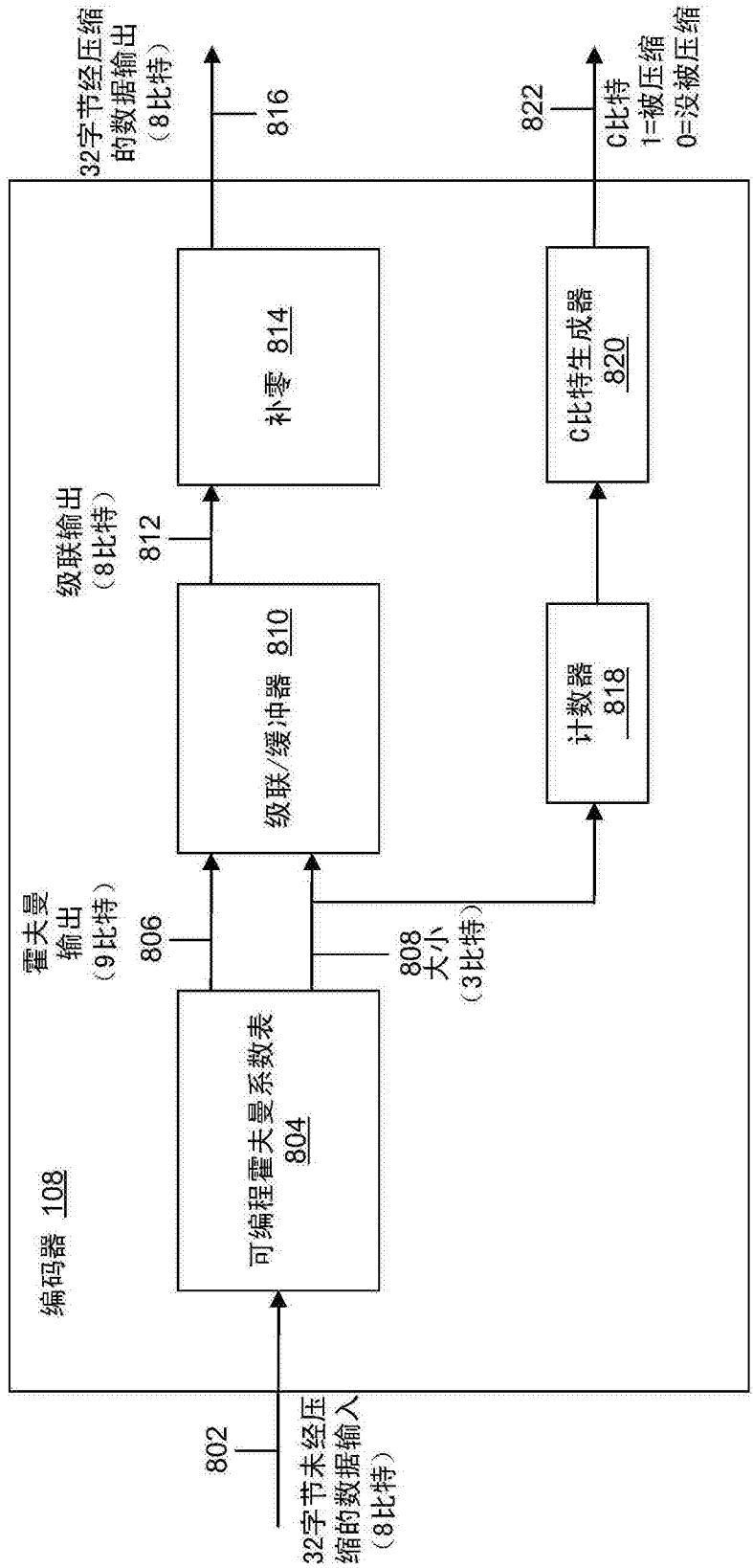


图8

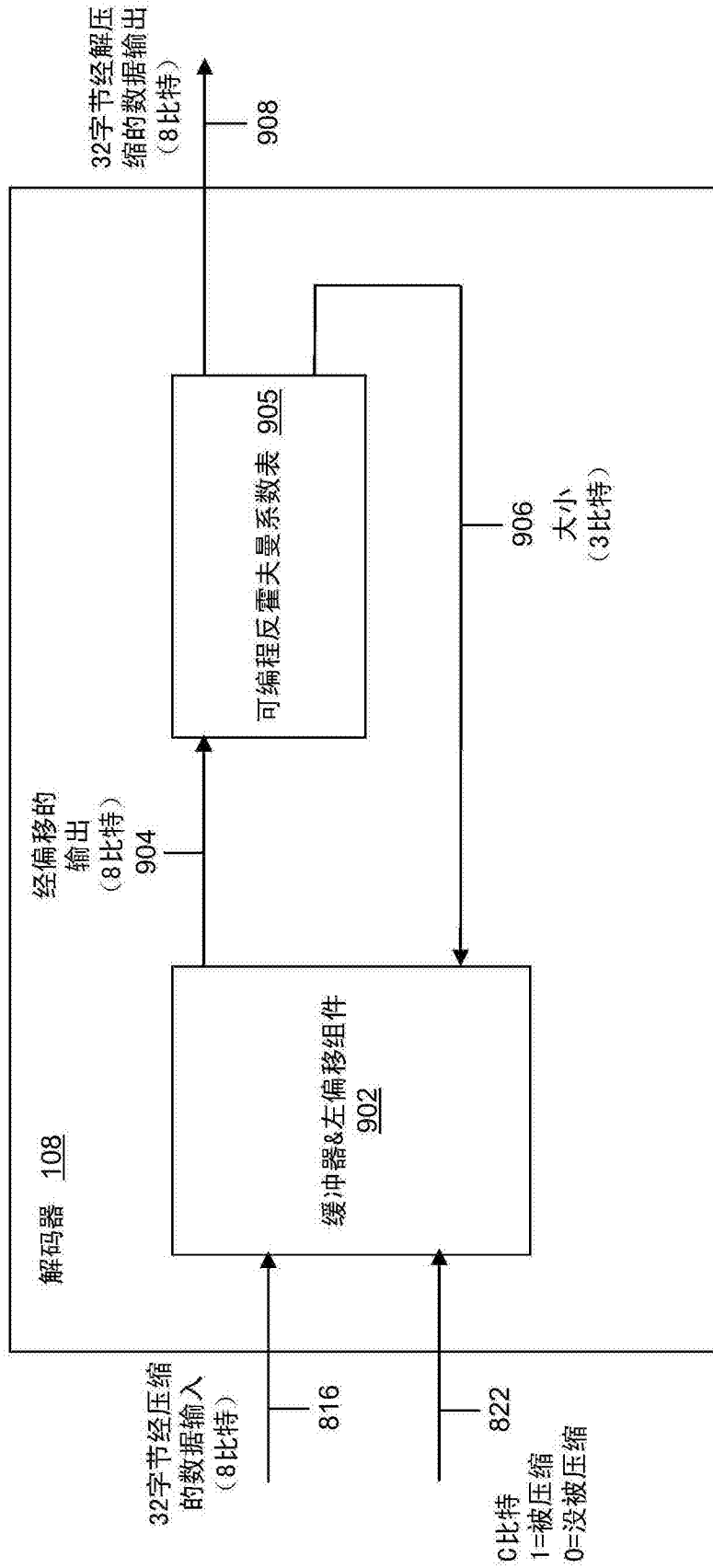


图9

大小	定义
0	未经压缩 (9比特输出)
1	经压缩的2比特
2	经压缩的3比特
3	经压缩的4比特
4	经压缩的5比特
5	经压缩的6比特
6	经压缩的7比特
7	经压缩的8比特

图10

客户端名称 1102	主ID 1104	压缩启用比特 1106	事务总数 1108	未经压缩的事务 1109 (C比特=0)
显示器处理器	22	1	65535	65535
GPU0	17	1	65535	32768

图11

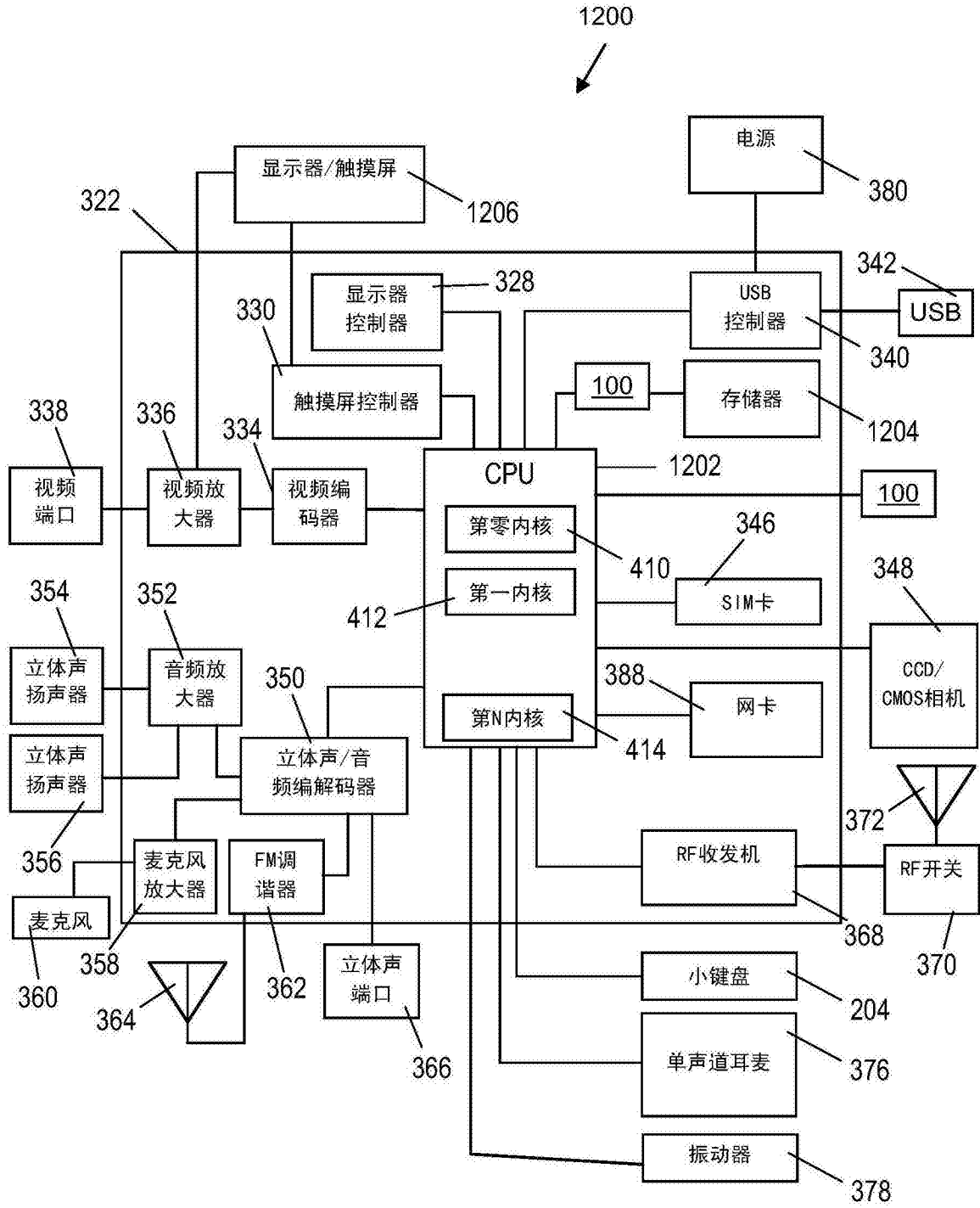


图12

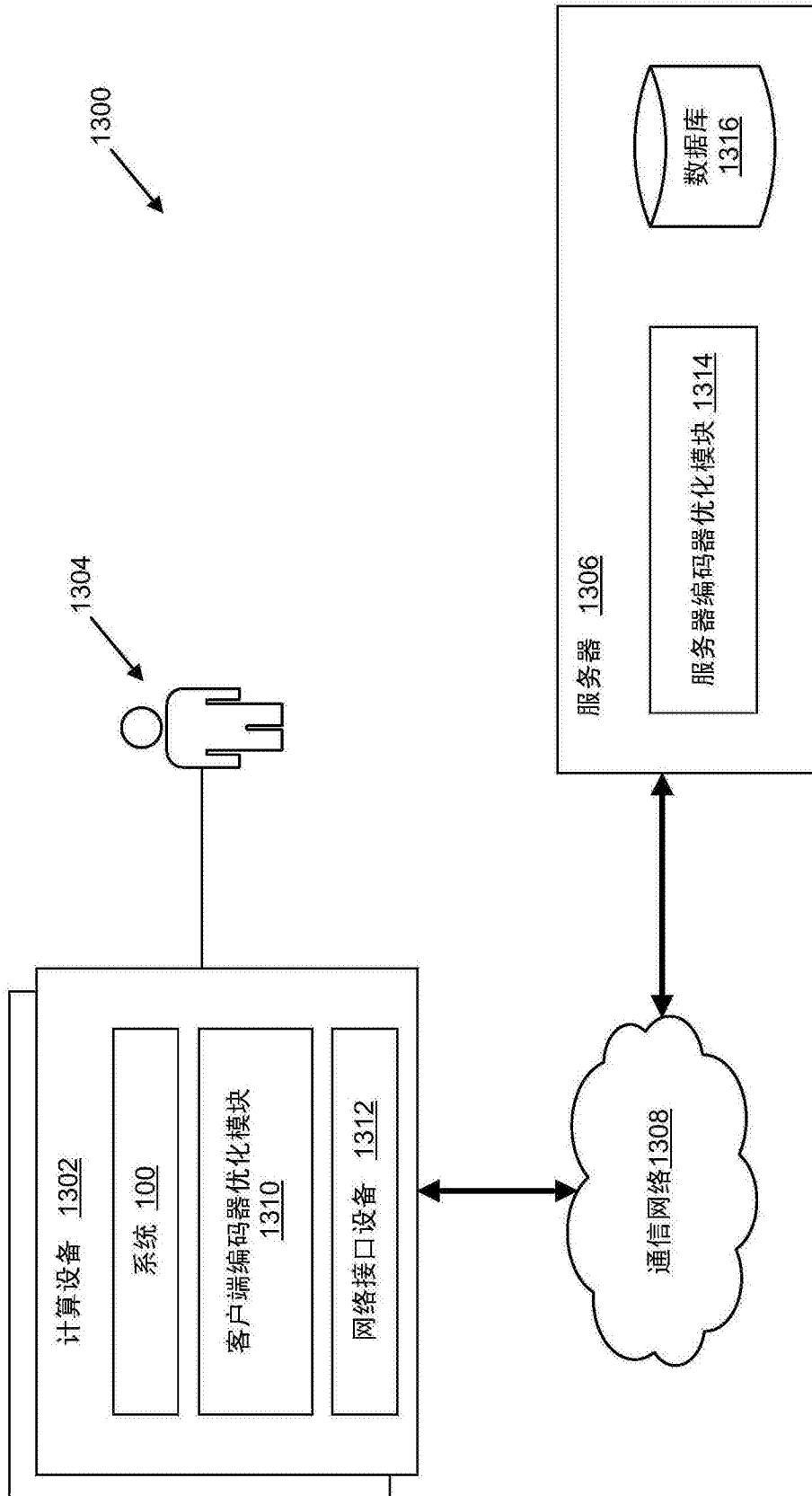


图13

1316
↙

用户ID <u>1402</u>	设备存储器图像 <u>1404</u>	码本 <u>1406</u>	码本压缩统计 <u>1407</u>	设备/用户度量 <u>1408</u>
↑ 1304a				
↑ 1304b				
↑ 1304c				
		▪ ▪ ▪		
↑ 1304n				

图14

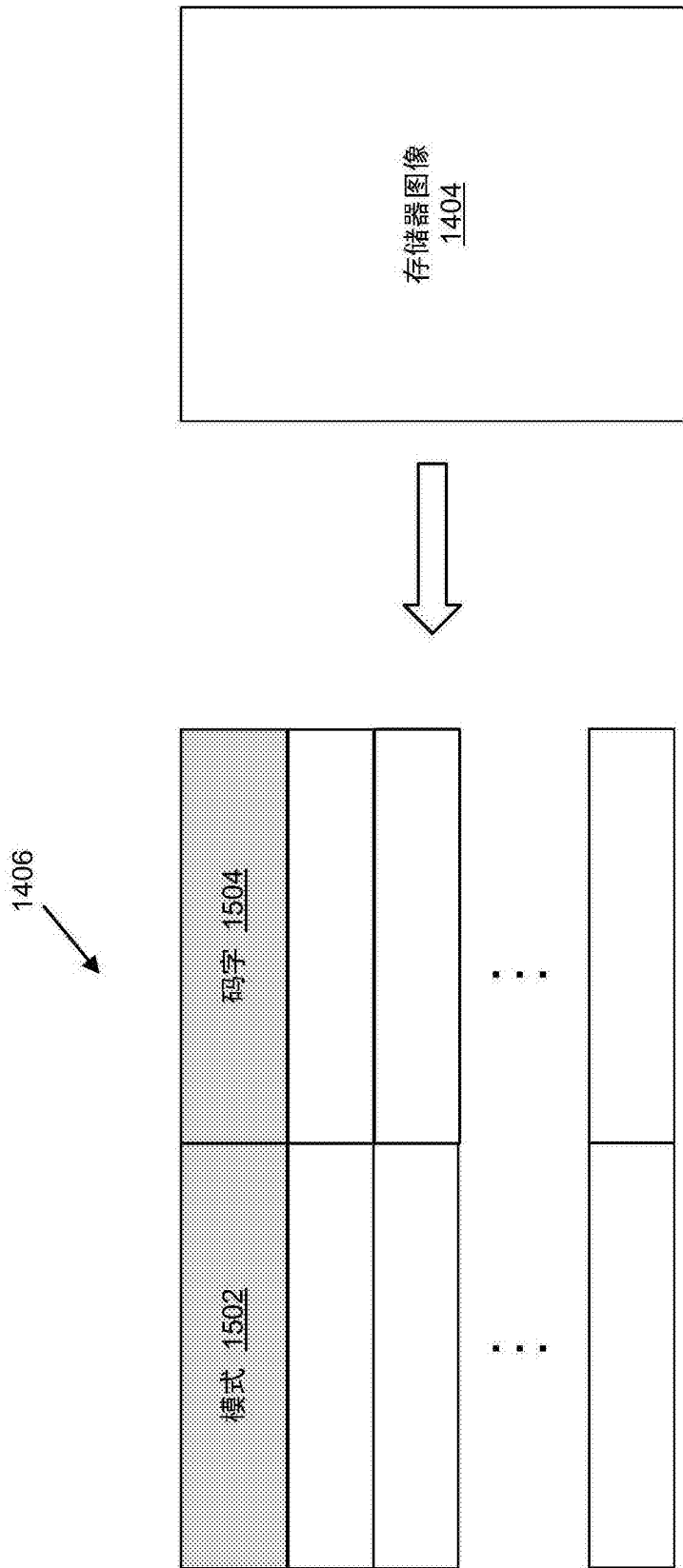


图15

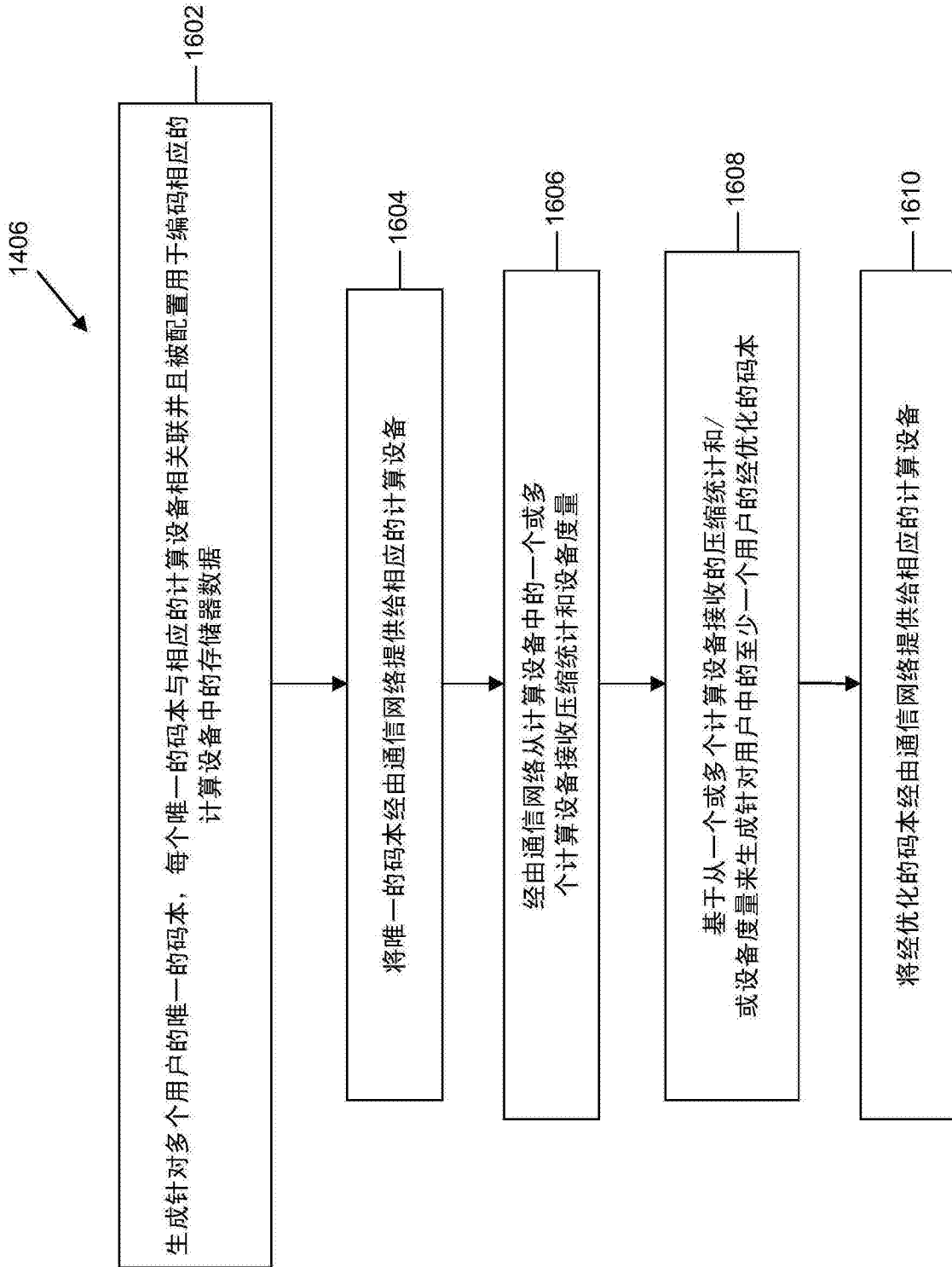


图16

1700
↙

度量 1702	值 1704	值 1706	值 1708
过程标识符	时间戳	%平均运行时间	版本信息
电话硬件标识符	硬件型号	电话修正	
CPU利用	时间戳	%平均利用	
C比特主ID	时间戳	%平均未经压缩	主ID名称
电话软件标识符	版本信息	版本信息	版本信息

图17