



(12) 发明专利申请

(10) 申请公布号 CN 103838679 A

(43) 申请公布日 2014. 06. 04

(21) 申请号 201210478383. 4

(22) 申请日 2012. 11. 22

(71) 申请人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术
产业园科技南路中兴通讯大厦法务部

(72) 发明人 张林生

(74) 专利代理机构 北京银龙知识产权代理有限
公司 11243

代理人 许静 安利霞

(51) Int. Cl.

G06F 12/08 (2006. 01)

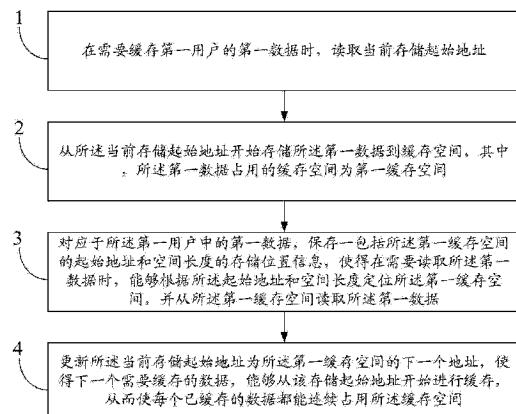
权利要求书2页 说明书6页 附图3页

(54) 发明名称

一种缓存处理方法及装置

(57) 摘要

本发明提供一种缓存处理方法及装置。其中方法包括：在需要缓存第一用户的第一数据时，读取当前存储起始地址；从当前存储起始地址开始存储第一数据到缓存空间，所述第一数据占用的缓存空间为第一缓存空间；对应于第一用户中的第一数据，保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息，使得在需要读取第一数据时，能够根据起始地址和空间长度定位所述第一缓存空间，并从第一缓存空间读取第一数据；更新当前存储起始地址为第一缓存空间的下一地址，使得下一个需要缓存的数据，能够从该存储起始地址开始进行缓存，从而使每个已缓存的数据都能连续占用所述缓存空间。本发明更好地利用缓存空间，减少为读取而保存的地址信息。



1. 一种缓存处理方法,其特征在于,包括如下步骤:

在需要缓存第一用户的第一数据时,读取当前存储起始地址;

从所述当前存储起始地址开始存储所述第一数据到缓存空间,其中,所述第一数据占用的缓存空间为第一缓存空间;

对应于所述第一用户中的第一数据,保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息,使得在需要读取所述第一数据时,能够根据所述起始地址和空间长度定位所述第一缓存空间,并从所述第一缓存空间读取所述第一数据;

更新所述当前存储起始地址为所述第一缓存空间的下一地址,使得下一个需要缓存的数据,能够从该存储起始地址开始进行缓存,从而使每个已缓存的数据都能连续占用所述缓存空间。

2. 根据权利要求 1 所述的缓存处理方法,其特征在于,从所述当前存储起始地址开始存储所述第一数据到缓存空间后还包括:

记录所述第一数据的空闲状态,从而根据所述空闲状态,判断所述第一数据是否被释放掉。

3. 根据权利要求 2 所述的缓存处理方法,其特征在于,还包括:

设置一阀值,所述阀值为存储一最大数据可能需要的缓存空间大小;

若在当前存储起始地址之后的剩余缓存空间小于预设阀值,则按照预设方法对所述缓存空间的数据进行整理,并在整理之后更新每个数据的起始地址;其中,所述预设方法为:在所述缓存空间中选择一最早存储的失效数据,在其起始地址,重新对该数据之后的有效数据连续存储,直至全部有效数据都已连续占用所述缓存空间;其中,所述失效数据为:已被释放掉的数据;所述有效数据为:未被释放掉的数据;

在整理完成后,更新所述当前存储起始地址为最后一个有效数据在所述缓存空间的下一地址。

4. 根据权利要求 3 所述的缓存处理方法,其特征在于,所述预设方法具体为:

在所述缓存空间中选择一最早存储的失效数据,通过其存储位置信息,获取该失效数据在所述缓存空间的起始地址,更新当前存储起始地址为所述失效数据的起始地址;

从该存储起始地址开始存储在所述失效数据之后的第一有效数据,更新第一有效数据的存储位置信息;

更新所述当前存储起始地址为所述第一有效数据所在缓存空间的下一地址,使得在其之后的下一个有效数据能够从该存储起始地址开始重新进行缓存,直至每个已缓存的有效数据都能连续占用所述缓存空间。

5. 根据权利要求 2 所述的存储缓存的方法,其特征在于,还包括:

对正被读取或者正被整理的有效数据进行锁定,使得一个有效数据在锁定过程中不能同时被读取与整理。

6. 一种缓存处理装置,其特征在于,包括:

读取存储起始地址模块,用于在需要缓存第一用户的第一数据时,读取当前存储起始地址;

缓存模块,用于从所述当前存储起始地址开始存储所述第一数据到缓存空间,其中,所述第一数据占用的缓存空间为第一缓存空间;

存储数据管理模块，用于对应于所述第一用户中的第一数据，保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息，使得在需要读取所述第一数据时，能够根据所述起始地址和空间长度定位所述第一缓存空间，并从所述第一缓存空间读取所述第一数据；

更新存储起始地址模块，用于更新所述当前存储起始地址为所述第一缓存空间的下一地址，使得下一个需要缓存的数据，能够从该存储起始地址开始进行缓存，从而使每个已缓存的数据都能连续占用所述缓存空间。

7. 根据权利要求 6 所述的缓存处理装置，其特征在于，所述存储数据管理模块还用于：记录所述第一数据的空闲状态，够根据所述空闲状态，判断所述第一数据是否被释放掉。

8. 根据权利要求 7 所述的缓存处理装置，其特征在于，还包括：

整理模块，用于设置一阀值，所述阀值为存储一最大数据可能需要缓存空间的大小；

若在当前存储起始地址之后的剩余缓存空间小于预设阀值，则按照预设方法对所述缓存空间的数据进行整理，并在整理之后更新每个数据的起始地址；其中，所述预设方法为：在所述缓存空间中选择一最早存储的失效数据，在其起始地址，重新对该数据之后的有效数据连续存储，直至全部有效数据都已连续占用所述缓存空间；其中，所述失效数据为：已被释放掉的数据；所述有效数据为：未被释放掉的数据；

在整理完成后，更新所述当前存储起始地址为最后一个有效数据在所述缓存空间的下一地址。

9. 根据权利要求 7 所述的缓存处理装置，其特征在于，还包括：

锁定模块，用于对正被读取或者正被整理的有效数据进行锁定，使得一个有效数据在锁定过程中不能同时被读取与整理。

一种缓存处理方法及装置

技术领域

[0001] 本发明涉及网络存储领域,特别是一种缓存处理方法及装置。

背景技术

[0002] SoC (System on Chip, 系统级芯片) 技术将 CPU/MPU (Micro Process Unit 微处理器)、DSP (Digital Signal Processor, 数字信息号处理)、硬件加速器、存储、外设接口等各个子系统,通过总线互联集成到一颗芯片上。这些子系统除了其内部数据处理所需要的本地专用存储 / 缓存之外,还往往由于各种各样的原因需要共享集中的存储 / 缓存,如各子系统相互之间需要数据交互缓冲;如处理多用户业务而共享存储。

[0003] 这些由多源 / 多用户共享的集中缓存可以在物理上是一块或几块存储单元,如一块 SRAM (Static Random-Access Memory, 静态存储器) 或一块 DRAM (Dynamic Random-Access Memory, 动态随机存储器) 或几块它们的任意组合,但我们可以逻辑上把它们看作是一块共享缓存。下文所述的共享缓存均基于逻辑上是一块存储单元来论述,而实际物理实现上可以是一块或多块存储单元的组合。

[0004] 如同任何公共资源一样,多源 / 多用户共享一块缓存的使用需要考虑存储资源的分配与管理。

[0005] 目前 SoC 设计中的缓存共享使用方法基本上是将共享缓存划分为 N 个相等大小的“片”,这个“片”的大小以能容纳下各源 / 各用户所需的最小数据为准,大的数据就占用整数个“片”。

[0006] 这种方法的缺点是需要很大的存储空间来记录每个“片”的使用状态及存储位置信息。特别是当系统所需的最小数据的长度很小时,相对于每个“片”用于存储的空间也会随之很小,在存储大数据时,需要用到很多的“片”。另外使用这方法还对共浪费缓存空间,如有些数据只比整数个“片”大一点点,这就需要再单独使用一个“片”进行存储。

发明内容

[0007] 本发明要解决的技术问题是提供一种能够更好的进行规划管理并能节约使用缓存空间的缓存处理方法及装置。

[0008] 为解决上述技术问题,本发明的实施例提供一种缓存处理方法,包括如下步骤:

[0009] 在需要缓存第一用户的第一数据时,读取当前存储起始地址;

[0010] 从所述当前存储起始地址开始存储所述第一数据到缓存空间,其中,所述第一数据占用的缓存空间为第一缓存空间;

[0011] 对应于所述第一用户中的第一数据,保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息,使得在需要读取所述第一数据时,能够根据所述起始地址和空间长度定位所述第一缓存空间,并从所述第一缓存空间读取所述第一数据;

[0012] 更新所述当前存储起始地址为所述第一缓存空间的下一地址,使得下一个需要缓存的数据,能够从该存储起始地址开始进行缓存,从而使每个已缓存的数据都能连续占用

所述缓存空间。

[0013] 其中,从所述当前存储起始地址开始存储所述第一数据到缓存空间后还包括:

[0014] 记录所述第一数据的空闲状态,从而根据所述空闲状态,判断所述第一数据是否被释放掉。

[0015] 其中,本方法还包括:

[0016] 设置一阀值,所述阀值为存储一最大数据可能需要的缓存空间大小;

[0017] 若在当前存储起始地址之后的剩余缓存空间小于预设阀值,则按照预设方法对所述缓存空间的数据进行整理,并在整理之后更新每个数据的起始地址;其中,所述预设方法为:在所述缓存空间中选择一最早存储的失效数据,在其起始地址,重新对该数据之后的有效数据连续存储,直至全部有效数据都已连续占用所述缓存空间;其中,所述失效数据为:已被释放掉的数据;所述有效数据为:未被释放掉的数据;

[0018] 在整理完成后,更新所述当前存储起始地址为最后一个有效数据在所述缓存空间的下一地址。

[0019] 其中,所述预设方法具体为:

[0020] 在所述缓存空间中选择一最早存储的失效数据,通过其存储位置信息,获取该失效数据在所述缓存空间的起始地址,更新当前存储起始地址为所述失效数据的起始地址;

[0021] 从该存储起始地址开始存储在所述失效数据之后的第一有效数据,更新第一有效数据的存储位置信息;

[0022] 更新所述当前存储起始地址为所述第一有效数据所在缓存空间的下一地址,使得在其之后的下一个有效数据能够从该存储起始地址开始重新进行缓存,直至每个已缓存的有效数据都能连续占用所述缓存空间。

[0023] 使得在其之后的下一个有效数据能够从该存储起始地址开始重新进行缓存,直至每个已缓存的有效数据都能连续占用所述缓存空间。

[0024] 其中,本方法还包括:

[0025] 对正被读取或者正被整理的有效数据进行锁定,使得一个有效数据在锁定过程中不能同时被读取与整理。

[0026] 本发明的实施例还提供一种缓存处理装置,包括:

[0027] 读取存储起始地址模块,用于在需要缓存第一用户的第一数据时,读取当前存储起始地址;

[0028] 缓存模块,用于从所述当前存储起始地址开始存储所述第一数据到缓存空间,其中,所述第一数据占用的缓存空间为第一缓存空间;

[0029] 存储数据管理模块,用于对应于所述第一用户中的第一数据,保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息,使得在需要读取所述第一数据时,能够根据所述起始地址和空间长度定位所述第一缓存空间,并从所述第一缓存空间读取所述第一数据;

[0030] 更新存储起始地址模块,用于更新所述当前存储起始地址为所述第一缓存空间的下一地址,使得下一个需要缓存的数据,能够从该存储起始地址开始进行缓存,从而使每个已缓存的数据都能连续占用所述缓存空间。

[0031] 其中,所述存储数据管理模块还用于:记录所述第一数据的空闲状态,根据所述空

闲状态，判断所述第一数据是否被释放掉。

[0032] 其中，本装置还包括：

[0033] 整理模块，用于设置一阀值，所述阀值为存储一最大数据可能需要缓存空间的大小；

[0034] 若在当前存储起始地址之后的剩余缓存空间小于预设阀值，则按照预设方法对所述缓存空间的数据进行整理，并在整理之后更新每个数据的起始地址；其中，所述预设方法为：在所述缓存空间中选择一最早存储的失效数据，在其起始地址，重新对该数据之后的有效数据连续存储，直至全部有效数据都已连续占用所述缓存空间；其中，所述失效数据为：已被释放掉的数据；所述有效数据为：未被释放掉的数据；

[0035] 在整理完成后，更新所述当前存储起始地址为最后一个有效数据在所述缓存空间的下一地址。

[0036] 其中，本装置还包括：

[0037] 锁定模块，用于对正被读取或者正被整理的有效数据进行锁定，使得一个有效数据在锁定过程中不能同时被读取与整理。

[0038] 本发明的上述技术方案的有益效果如下：

[0039] 将数据连续存储在缓存空间中，相对于现有技术将缓存空间分成若干个片段而言，可以更好的利用缓存空间；进一步地，现有技术中，为了实现数据的读取，需要记录这些片段的缓存地址，信息存储量极为庞大，而本实施例的缓存空间由于连续存储数据，只需要记录每个数据的所在缓存空间中的起始地址以及空间长度就能实现读取，其信息存储量取决于数据包的数量。

附图说明

[0040] 图 1 为本发明的缓存处理方法的示意图；

[0041] 图 2 为本发明的缓存处理方法的流程图；

[0042] 图 3 为本发明的缓存处理方法中的对缓存空间进行整理的流程图；

[0043] 图 4 为本发明的缓存处理装置的示意图。

具体实施方式

[0044] 为使本发明要解决的技术问题、技术方案和优点更加清楚，下面将结合附图及具体实施例进行详细描述。

[0045] 实施例 1

[0046] 如图 1 所示，一种缓存处理方法，包括如下步骤：

[0047] 步骤 1，在需要缓存第一用户的第一数据时，读取当前存储起始地址；

[0048] 步骤 2，从所述当前存储起始地址开始存储所述第一数据到缓存空间，其中，所述第一数据占用的缓存空间为第一缓存空间；

[0049] 步骤 3，对应于所述第一用户中的第一数据，保存一包括所述第一缓存空间的起始地址和空间长度的存储位置信息，使得在需要读取所述第一数据时，能够根据所述起始地址和空间长度定位所述第一缓存空间，并从所述第一缓存空间读取所述第一数据；

[0050] 步骤 4，更新所述当前存储起始地址为所述第一缓存空间的下一地址，使得下一个

需要缓存的数据，能够从该存储起始地址开始进行缓存，从而使每个已缓存的数据都能连续占用所述缓存空间。

[0051] 本实施例的方法将数据连续存储在缓存空间中，相对于现有技术将缓存空间分成若干个片段而言，可以更好的利用缓存空间；进一步地，现有技术中，为了实现数据的读取，需要记录这些片段的缓存地址，信息存储量极为庞大，而本实施例的缓存空间由于连续存储数据，只需要记录每个数据的所在缓存空间中的起始地址以及空间长度就能实现读取，其信息存储量取决于数据包的数量。

[0052] 下面对图 1 所示方法详细描述：

[0053] 如图 2 所示，在需要缓存第一用户的第一数据时进行一下步骤：

[0054] 步骤 S201，获取当前存储起始地址 st_add.1；

[0055] 步骤 S202，从 st_add.1 开始，将第一用户的第一数据存储进缓存空间，第一数据占用的缓存空间为第一缓存空间；

[0056] 步骤 S203，在额外存储空间中保存第一缓存空间的起始地址（如图中 2 所示的“st_add.1”）、空间长度（如图中 2 所示的“len.1”）以及第一数据的空闲状态（如图中 2 所示的“valid”，valid=1 表示该数据未被释放，valid=0 表示数据已被释放掉），使得在需要读取第一数据时，只需找到第一缓存空间的起始地址 st_add.1 以及空间长度为 len.1 即可定位第一缓存空间，读取第一数据；

[0057] 为了使每个存储的数据都能连续性占用缓存空间，在步骤 S203 完成之后，进行步骤 S204，更新所述当前存储起始为第一缓存空间的下一地址；

[0058] 此时的存储起始地址应为 st_add.2，当需要继续存储数据时，如存储第二数据，可进行一下步骤：

[0059] 步骤 S205，从 st_add.2 开始，将第二数据存储进缓存空间中，第二数据占用的缓存空间为第二缓存空间（如图 2 所示）；

[0060] 需要指出的是，所述第二数据并不一定是第一用户需要缓存的数据，也可以属于其他用户；

[0061] 步骤 S206，在额外存储空间中保存第二缓存空间的起始地址（如图中 2 所示的“st_add.2”）、空间长度（如图中 2 所示的“len.2”）以及第二数据的空闲状态（如图中 2 所示的“valid”）；

[0062] 步骤 S207，更新所述当前存储起始为第二缓存空间的下一地址。

[0063] 之后其他需要存储的数据都应按照上述方法保存进缓存空间中，使得每个已缓存的数据都能连续占用缓存空间，并在额外存储空间中记录每个数据的起始地址、空间长度。

[0064] 由于缓存空间有限，当存储某一数据时，可能会出现存储起始地址之后的剩余缓存空间不足以再进行存储，而有些缓存数据已经被读取并释放，其对应的缓存空间也已清空，这就需要对整个缓存空间进行整理，将已被清空的缓存空间整合到当前存储初始地址之后的剩余缓存空间中，其方法为：

[0065] 设置一阀值，所述阀值为存储一最大数据可能需要的缓存空间大小；

[0066] 若在当前存储起始地址之后的剩余缓存空间小于预设阀值，则按照预设方法对所述缓存空间的数据进行整理，并在整理之后更新每个数据的起始地址；其中，所述预设方法为：在所述缓存空间中选择一最早存储的失效数据，在其起始地址，重新对该数据之后的有

效数据连续存储,直至全部有效数据都已连续占用所述缓存空间;其中,所述失效数据为:已被释放掉的数据;所述有效数据为:未被释放掉的数据;

[0067] 下面对预设方法进行详细描述:

[0068] 如图3所示,缓存空间中已存有6个数据,通过额外缓存空间记录的空闲状态可知数据2与数据5为失效数据(即空闲状态Valid=0),都已被释放掉,其所在的缓存空间也被清空。数据1、3、4、6为有效数据(即空闲状态Valid=1),在缓存空间中未被释放。

[0069] 当接到指令存储数据7时,预设一个阈值,该阈值为存储一个最大数据所需要的空间大小,若剩余的缓存空间小于预设阀值,则说明目前未被使用缓冲空间无法存储数据7,需要对缓冲空间进行整理,整理步骤为:

[0070] 步骤S1,更新当前存储起始地址为数据2的起始地址;将数据3在数据2的起始地址开始重新存储(即将数据3所对应的缓存空间向上挪一段距离,其距离等于数据2的空间长度),并更新额外存储空间中记录数据2的起始地址;

[0071] 步骤S2,更新当前存储起始地址为数据3所在缓存空间的下一地址,在该存储起始地址重新存储数据4,并更新额外存储空间中数据4的起始地址;

[0072] 步骤S3,更新当前存储起始地址为数据4所在缓存空间的下一地址,在该存储起始地址重新存储数据6,从而使数据2与数据5所对应的缓存空间整合到当前存储起始地址之后的剩余缓存空间中;之后更新额外存储空间中数据6的起始地址;

[0073] 步骤S4,在缓存空间整理完成之后,存储数据7。需要指出的是,所有有效数据都应是连续占用缓存空间。

[0074] 另外当需要对目标数据进行读取时,可能出现目标数据正被进行整理,其在缓存空间的位置因为发生变化致使读取出错。为了解决这一问题,可以对正被读取或者正被整理的有效数据进行锁定,在锁定过程中使得一个有效数据不能同时被读取与整理。例如:在数据3被整理的时候,可将其锁定,直到整理完成后再解除锁定,在数据3被锁定时,无法接收读取请求,从而避免整理过程干扰读取过程。

[0075] 本实施例的方法有益效果如下:

[0076] 1. 将每个数据连续存储在缓存空间中,可以更好的利用缓存空间,节约资源;

[0077] 2. 对缓存空间进行整理,将已被清空的缓存空间整合到当前存储初始地址之后的剩余缓存空间中,保持全部有效数据都已连续占用所述缓存空间,可以在有限缓存空间中存储更多的数据,进一步节约资源。

[0078] 3. 现有技术中,将整个缓存空间分成若干个片段,有些大数据会由很多个片段来保存,为了实现数据的读取,需要记录这些片段的缓存地址,信息存储量极为庞大,而本实施例的由于缓存空间连续存储数据,只需要记录每个有效数据所在缓存空间中的起始地址以及空间长度就能实现读取。

[0079] 实施例2

[0080] 一种缓存处理装置,包括:

[0081] 读取存储起始地址模块,用于在需要缓存第一用户的第一数据时,读取当前存储起始地址;缓存模块,用于从所述当前存储起始地址开始存储所述第一数据到缓存空间,其中,所述第一数据占用的缓存空间为第一缓存空间;

[0082] 存储数据管理模块,用于对应于所述第一用户中的第一数据,保存一包括所述第

一缓存空间的起始地址和空间长度的存储位置信息,使得在需要读取所述第一数据时,能够根据所述起始地址和空间长度定位所述第一缓存空间,并从所述第一缓存空间读取所述第一数据;另外,存储数据管理模块还可以记录所述第一数据的空闲状态,并根据所述空闲状态,判断所述第一数据是否被释放掉;

[0083] 更新存储起始地址模块,用于更新所述当前存储起始地址为所述第一缓存空间的下一地址,使得下一个需要缓存的数据,能够从该存储起始地址开始进行缓存,从而使每个已缓存的数据都能连续占用所述缓存空间。

[0084] 由于缓存空间有限,当存储某一数据时,可能会出现存储起始地址之后的剩余缓存空间不足以再进行存储,而有些缓存的数据已经被读取并释放,其对应的缓存空间也已清空,这就需要对整个缓存空间进行整理。

[0085] 整理模块会设置一阀值,所述阀值为存储一最大数据可能需要缓存空间的大小;若在当前存储起始地址之后的剩余缓存空间小于预设阀值,则按照预设方法对所述缓存空间的数据进行整理,并在整理之后更新每个数据的起始地址;其中,所述预设方法为:在所述缓存空间中选择一最早存储的失效数据,在其起始地址,重新对该数据之后的有效数据连续存储,直至全部有效数据都已连续占用所述缓存空间;其中,所述失效数据为:已被释放掉的数据;所述有效数据为:未被释放掉的数据;在整理完成后,更新所述当前存储起始地址为最后一个有效数据在所述缓存空间的下一地址。

[0086] 所述预设方法具体为:在所述缓存空间中选择一最早存储的失效数据,通过其存储位置信息,获取该失效数据在所述缓存空间的起始地址,更新当前存储起始地址为所述失效数据的起始地址;从该存储起始地址开始存储在所述失效数据之后的第一有效数据,更新第一有效数据的存储位置信息;更新所述当前存储起始地址为所述第一有效数据所在缓存空间的下一地址,使得在其之后的下一个有效数据能够从该存储起始地址开始重新进行缓存,直至每个已缓存的有效数据都能连续占用所述缓存空间。

[0087] 为了避免整理模块在进行整理过程时,干扰缓存空间中数据的读取,本装置还包括一个锁定模块,可以对正被读取或者正被整理的有效数据进行锁定,使得一个有效数据在锁定过程中不能同时被读取与整理。

[0088] 本实施例的方法有益效果如下:

[0089] 1. 将每个数据连续存储在缓存空间中,可以更好的利用缓存空间,节约资源;

[0090] 2. 对缓存空间进行整理,将已被清空的缓存空间整合到当前存储初始地址之后的剩余缓存空间中,保持全部有效数据都已连续占用所述缓存空间,可以在有限缓存空间中存储更多的数据,进一步节约资源。

[0091] 3. 现有技术中,将整个缓存空间分成若干个片段,有些大数据会由很多个片段来保存,为了实现数据的读取,需要记录这些片段的缓存地址,信息存储量极为庞大,而本实施例的由于缓存空间连续存储数据,只需要记录每个有效数据的所在缓存空间中的起始地址以及空间长度就能实现读取。

[0092] 以上所述是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明所述原理的前提下,还可以作出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

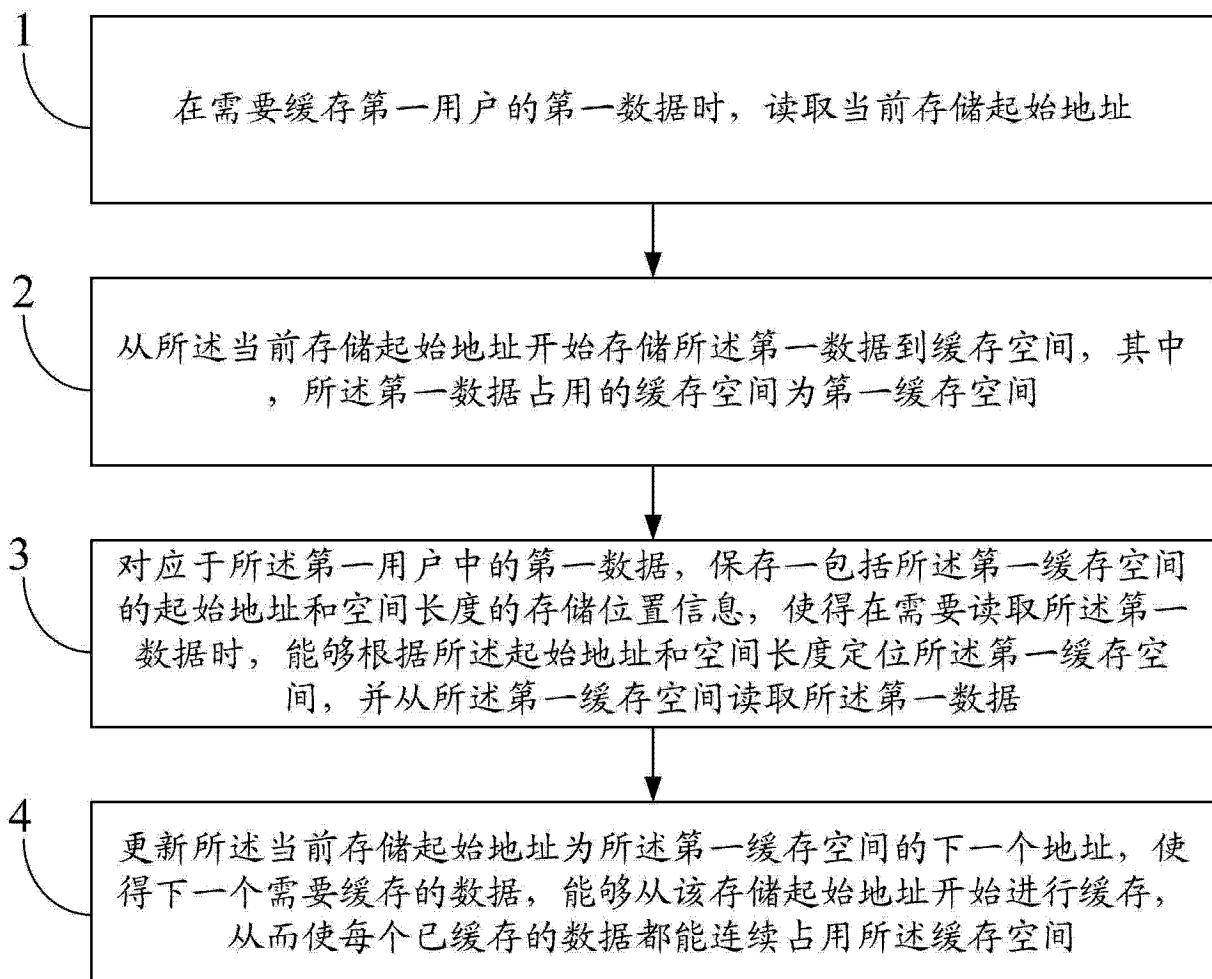


图 1

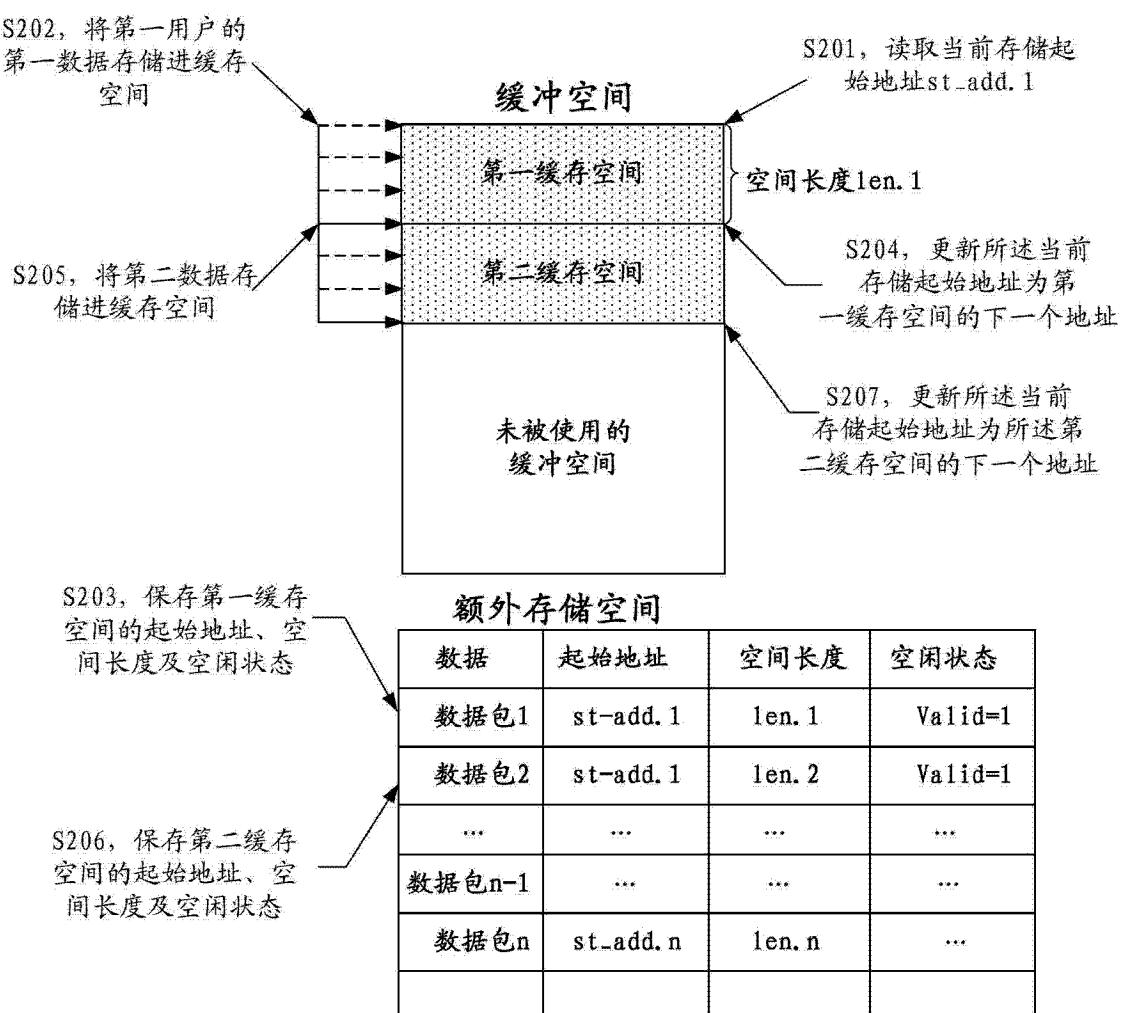


图 2

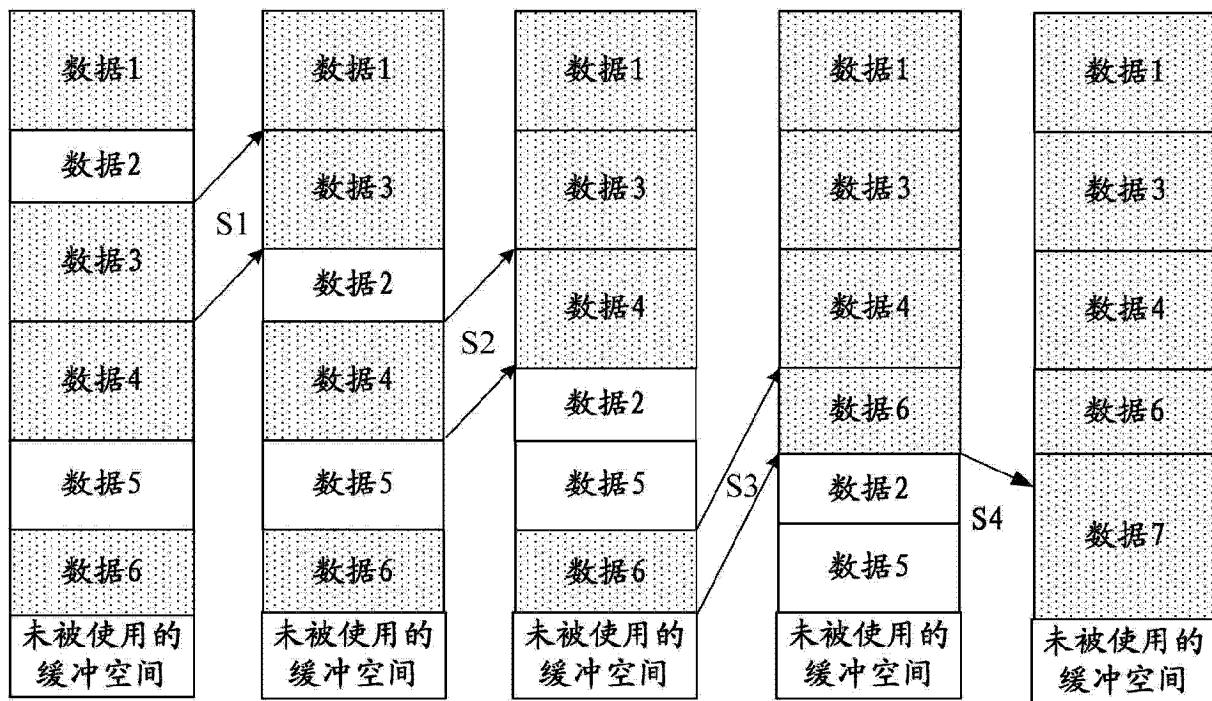


图 3

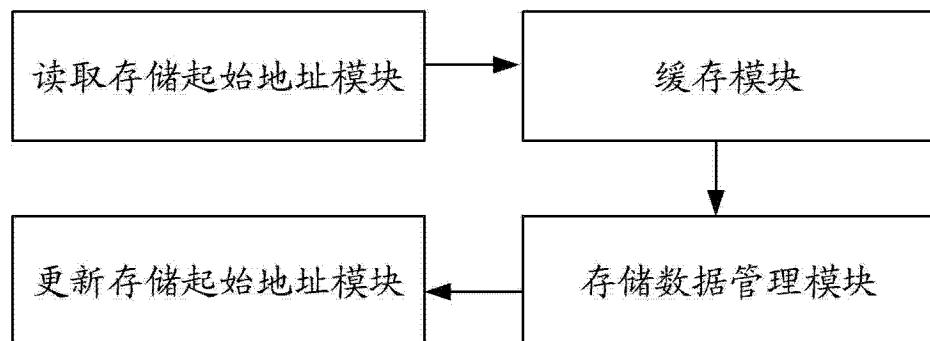


图 4