

## (12) 发明专利

(10) 授权公告号 CN 101114376 B

(45) 授权公告日 2010. 06. 23

(21) 申请号 200710123327.8

US 6288730 B1, 2001. 09. 11, 说明书第 8 栏

(22) 申请日 2007. 06. 20

第 49 行 - 第 9 栏第 2 行, 第 9 栏第 22-27, 31-33, 43-45 行, 第 10 样第 3-5, 57-61 行, 第 11 样第 49-62 行, 第 13 样第 46-51 行、图 2-5, 11, 13A-B.

(30) 优先权数据

60/815, 053 2006. 06. 20 US

审查员 许菲菲

(73) 专利权人 威盛电子股份有限公司

地址 中国台湾台北县

(72) 发明人 徐建明 陈文中 李亮

(74) 专利代理机构 北京林达刘知识产权代理事务所 11277

代理人 刘新宇

(51) Int. Cl.

G06T 1/60 (2006. 01)

G06F 12/08 (2006. 01)

(56) 对比文件

JP 2005-122479 A, 2005. 05. 12, 全文 .

US 7053904 B1, 2006. 05. 30, 全文 .

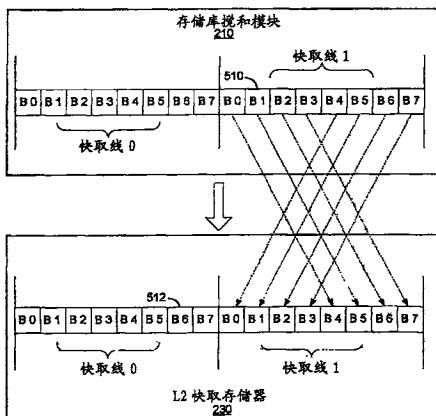
权利要求书 2 页 说明书 10 页 附图 8 页

(54) 发明名称

减少存储库碰撞的方法及处理图形的计算机系统

(57) 摘要

本发明提供一种减少存储库碰撞的方法及处理图形的计算机系统。一实施例包括一种减少 2 级快取存储器内的存储库碰撞的方法，方法包含以下步骤：自用以储存用于图形处理单元内的纹理过滤的纹理数据之外部存储器读取纹理数据；将纹理数据分为存储库；每隔一快取线即对存储库执行存储库搅和操作；以及将数据的存储库写入至 2 级快取存储器。本发明所述的减少存储库碰撞的方法及处理图形的计算机系统，可用于有效地管理并组织纹理数据，以使纹理过滤器处理时所需的任何纹理数据在图形处理系统中可易于撷取。



1. 一种减少存储库碰撞的方法,用于减少 2 级快取存储器内的存储库碰撞,其特征在于,包含:

自用以储存用于一图形处理单元内的纹理过滤操作的纹理数据的一外部存储器接收纹理数据;

将所述纹理数据分为多个存储库,所述存储库储存在至少一快取线中;

每隔一快取线即对所述存储库执行存储库搅和操作;以及

将所述纹理数据的所述存储库写入至所述 2 级快取存储器;

其中将所述纹理数据分为多个存储库以及将所述纹理数据的所述存储库写入至所述 2 级快取存储器的步骤是在逐存储库的基础上执行;

其中所述存储库搅和操作是指重新映射位于所述快取线中的所述存储库。

2. 根据权利要求 1 所述的减少存储库碰撞的方法,其特征在于,将所述纹理数据分为多个存储库包含:

将所述纹理数据组织成具有对应索引的连续存储库,其中所述存储库储存在至少一快取线中;以及

将所述存储库组织成快取线。

3. 根据权利要求 1 所述的减少存储库碰撞的方法,其特征在于,重新映射包含:

将存储库 0 与存储库 4 调换;

将存储库 1 与存储库 5 调换;

将存储库 2 与存储库 6 调换;以及

将存储库 3 与存储库 7 调换。

4. 根据权利要求 3 所述的减少存储库碰撞的方法,其特征在于,重新映射更包含:

分析待重新映射的所述存储库以确定新存储库碰撞是否将由于重新映射所述存储库而出现;以及

利用不同的存储库搅和模式,以避免新存储库碰撞。

5. 一种处理图形的计算机系统,其特征在于,包含:

一外部存储器,用以储存用于一图形处理单元内的纹理过滤操作的纹理数据;

一存储库搅和模块,用以自所述外部存储器接收纹理数据,将所述纹理数据分为多个存储库,所述存储库储存在至少一快取线中,且每隔一快取线即执行存储库搅和操作以重新映射所述纹理数据的所述存储库;以及

一 2 级快取存储器,耦接至所述存储库搅和模块,其中所述 2 级快取存储器可用以储存所述经重新映射的纹理数据。

6. 根据权利要求 5 所述的处理图形的计算机系统,其特征在于,所述存储库搅和模块用于下列情况其中之一:

仅在当自所述 2 级快取存储器提取纹理数据出现存储库碰撞时,执行所述存储库搅和操作;以及

在自外部存储器接收所述纹理数据之后执行所述存储库搅和操作。

7. 根据权利要求 5 所述的处理图形的计算机系统,其特征在于,所述存储库搅和模块根据一映射机制执行所述存储库搅和操作,所述映射机制包含:

将存储库 0 与存储库 4 调换;

将存储库 1 与存储库 5 调换；  
将存储库 2 与存储库 6 调换；以及  
将存储库 3 与存储库 7 调换。

8. 根据权利要求 7 所述的处理图形的计算机系统，其特征在于，所述存储库搅和模块可用以分析待执行的所述存储库搅和操作，以确定新存储库碰撞是否将由于利用所述映射机制而出现，且其中所述存储库搅和模块更可用以利用不同的存储库搅和模式以避免新的存储库碰撞。

9. 一种减少存储库碰撞的方法，用于减少 2 级快取存储器内的存储库碰撞，其特征在于，包含：

自所述 2 级快取存储器提取纹理数据，其中所述纹理数据用于一图形处理单元内的纹理过滤；

确定在提取所述纹理数据时是否已出现存储库碰撞，其中所述纹理数据分为位于所述 2 级快取存储器内的至少一快取线中的多个存储库；以及

若出现所述存储库碰撞，则每隔一快取线对所述存储库执行存储库搅和操作；

其中所述纹理数据在逐存储库的基础上被分为多个存储库，执行所述存储库搅和操作是指根据所述快取线重新映射涉及所述存储库碰撞的所述存储库。

10. 根据权利要求 9 所述的减少存储库碰撞的方法，其特征在于，所述重新映射包含：

将存储库 0 与存储库 4 调换；  
将存储库 1 与存储库 5 调换；  
将存储库 2 与存储库 6 调换；以及  
将存储库 3 与存储库 7 调换。

11. 根据权利要求 10 所述的减少存储库碰撞的方法，其特征在于，所述重新映射更包含：

分析待重新映射的所述存储库以确定新存储库碰撞是否将由于重新映射所述存储库而出现；以及

利用不同的存储库搅和模式以避免新的存储库碰撞。

## 减少存储库碰撞的方法及处理图形的计算机系统

### 技术领域

[0001] 本发明是关于图形处理,且特别是关于用于图形数据管理的系统以及方法。

### 背景技术

[0002] 如已知的先前技术,计算机图形处理系统需处理大量数据,其中包括纹理(texture)数据。纹理为具有一(u,v)坐标空间的数字影像(通常为矩形的)。纹理的最小可定址单元为纹理元素(texel),根据其位置而被指派一特定(u,v)坐标。在纹理映射(mapping)运算中,纹理被映射至图解模型的表面来呈现模型以建立目的影像。在目的影像中,纹理元素位于(x,y)坐标系统中的特定坐标处。纹理映射的目的为在物件的表面上提供实际外观。

[0003] 纹理数据通常内存在系统存储器中,系统存储器为共用资源。在许多计算机系统中,其他元件可能尝试存取由图形处理系统使用的数据或利用共用的系统总线,此两种情形可能会导致图形处理系统的数据存取时间增加。此外,请求来自系统存储器的数据由于其他原因可能会用去过多的时间。因此,存取系统存储器可能会对图形处理系统具有效能抑制效应。

[0004] 在现代计算机图形中,纹理处理可包括许多不同类型的纹理映射技术,包括立方映射(cubic mapping)、mip映射(mipmapping),以及各向异性(anisotropic)映射。由于执行纹理处理所需的算术以及逻辑运算十分复杂以及大量,使得除了用于其他图形处理运算的电路之外,纹理处理专用电路的使用成为必要。因此,此等纹理处理专用电路可包括显著百分比的构成图形处理单元的硬件门(gate)。

[0005] 纹理映射在计算与存储器存取上需付出昂贵的代价。改良数据可存取性的一种技术为经由专用于储存纹理数据的纹理快取存储器,在数据需要进行纹理处理之前,将纹理数据自系统存储器提供给纹理快取存储器,借此提供图形系统所需的纹理数据,且减少对存取系统存储器的需要。此方式又可减少与存储器延迟(latency)相关的问题。然而,由于可能偶尔出现存储库碰撞(bank collision)或快取存储器猛移(cache thrashing),即使使用纹理快取存储器,存储器延迟仍继续成为问题。因此,产业中存在对纹理像砖(tile)数据于快取存储器储存中改良的需要。

### 发明内容

[0006] 本发明提供用于储存图形数据的系统以及方法。一实施例包括用于减少2级(L2)快取存储器内的存储库碰撞的方法,其包含:自用以储存图形处理单元内执行纹理过滤所需的纹理数据之外部存储器读取纹理数据;将纹理数据分为存储库,所述存储库储存在至少一快取线中;每隔一快取线即对存储库执行存储库搅和操作(bank swizzle operation);以及将数据的存储库写入至2级快取存储器;其中将所述纹理数据分为多个存储库以及将所述纹理数据的所述存储库写入至所述2级快取存储器的步骤是在逐存储库的基础上执行;其中所述存储库搅和操作是指重新映射位于所述快取线中的所述存储

库。

[0007] 另一实施例包括用于处理图形数据的计算机系统,此计算机系统包含:外部存储器,用以储存用于图形处理单元内的纹理过滤的纹理数据;存储库搅和模块,用以自外部存储器接收纹理数据,将纹理数据分为存储库,所述存储库储存在至少一快取线中,且每隔一快取线即对纹理数据的存储库执行存储库搅和操作;以及2级(L2)快取存储器,其耦接至存储库搅和模块,其中2级快取存储器可用以储存经搅和过的纹理数据。

[0008] 又一实施例包括用于减少2级快取存储器内的存储库碰撞的方法,方法包含:自2级快取存储器提取纹理数据,此纹理数据用于图形处理单元内的纹理过滤;确认在提取纹理数据时是否已出现存储库碰撞,其中纹理数据被分为位于2级快取存储器内的至少一快取线中的多个存储库;以及在会出现存储库碰撞时,每隔一快取线对存储库执行存储库搅和操作;其中所述纹理数据在逐存储库的基础上被分为多个存储库,执行所述存储库搅和操作是指根据所述快取线重新映射涉及所述存储库碰撞的所述存储库。

[0009] 本发明所述的减少存储库碰撞的方法及处理图形的计算机系统,可用于有效地管理并组织纹理数据,以使纹理过滤器处理时所需的任何纹理数据在图形处理系统中可易于撷取。

## 附图说明

- [0010] 图1为图形处理器系统的实施例的方块图。
- [0011] 图2说明图形处理系统内的纹理快取系统的部分基本组件。
- [0012] 图3为例L2快取存储器组织格式的方块图。
- [0013] 图4绘示经由存储库搅和模块自外部存储器转送纹理数据至L2快取存储器的实施例。
- [0014] 图5为存储库搅和操作的实施例。
- [0015] 图6为具有32BPE的解析度的纹理数据的4×8占据面积。
- [0016] 图7为存储库搅和操作之后邻近快取线中同样编号存储库之间的距离。
- [0017] 图8为存储库搅和操作之前以及之后占据面积的大小。
- [0018] 图9绘示用于执行存储库搅和操作以减少存储库碰撞的方法的实施例流程图。
- [0019] 图10绘示用于执行存储库搅和操作以减少存储库碰撞的方法的另一实施例的流程图。

## 具体实施方式

[0020] 在检查以下图式以及实施方式时,本发明的其他系统、方法、特征以及益处将对于本领域技术人员为显而易见的或变得显而易见。所有此等额外系统、方法、特征以及益处是打算被包括在此描述内,在本发明的范畴内,且通过随附权利要求书加以保护。

[0021] 在概括了本发明的各种观点之后,现将详细参看如图式中所说明的本发明的描述。虽然将结合此等图式来描述本发明,但是不意欲将本发明限制于本文所揭露的一或多个实施例。而是意欲涵盖包括在如通过随附权利要求书界定的本发明的精神以及范畴内的所有替代、修改以及均等物。

[0022] 本发明所揭露的系统以及方法可用于有效地管理并组织纹理数据,以使(例如)

纹理过滤器处理时所需的任何纹理数据在图形处理系统中可易于撷取。通常，自系统存储器接收众多不同格式以及大小的纹理数据，若未考虑此数据的大小或组成方式可能会导致纹理快取存储器资源的使用上较无效率。此种无效率的导因之一为在撷取纹理数据时存在过多延迟。可理解地，撷取纹理像砖数据 (texture tiling data) 的延迟导致在处理此数据中的延迟。在某些情况下，此延迟可能通过出现在快取存储器内的存储库相互碰撞而引起。当试图自具有相同索引且储存在不同快取线中的存储库撷取数据时，则出现存储库碰撞。给定纹理数据的定域性 (localized nature)，紧接地储存在一起的存储库将更可能在存储器存取期间一起受到提取。

[0023] 在作图形处理时，纹理数据不必要一次一快取线地撷取，而是在逐存储库的基础上撷取。因此，在存储器存取期间可自任何给定快取线撷取存储库的任何组合。举例而言，在存储器存取期间，可一起提取（快取线 1，存储库 0）、（快取线 2，存储库 1）等。然而，若试图同时撷取（快取线 1，存储库 0）以及（快取线 2，存储库 0），则出现存储库碰撞，因此具有相同索引的两个存储库必须独立地加以撷取，导致至少一周期的延迟。因此，由于存储库碰撞的缘故，需要额外周期来撷取存储库。用于执行存储库搅和 (swizzle) 操作的系统以及方法的实施例可通过增加具有相同索引数字且位于邻近快取线中的存储库相互间的距离来减少存储库碰撞的可能性。

[0024] 简要参看图 6 以及图 7，其说明具有相同索引数字且位于邻近快取线中的存储库之间的距离。如图 6 中所揭示，（快取线 0，存储库 0）以及（快取线 1，存储库 0）是以距离 A 隔开，其中通过纹理元素来量测距离。对快取线 1 执行存储库搅和操作可有效地将此等两个存储库之间的距离增加至新距离 A，如图 7 中所示。因此，本发明揭露一种系统以及方法可用于对纹理数据执行存储库搅和操作以便减少存储库碰撞的可能性，借此导致在计算机图形环境中提取数据耗费较少时脉周期。

[0025] 现参看图 1，其为计算机系统 100 的简化方块图。计算机系统 100 包括 CPU 102、系统存储器 104 以及图形处理系统 110。CPU 102 执行各种功能，包括确定产生图形显示所需的信息（诸如，观察点位置）。系统存储器 104 储存各种数据，包括诸如纹理数据 106 的图形显示数据。根据 CPU 102 所确定的信息以及储存在系统存储器 104 中的数据，图形处理系统 110 产生用于显示元件 130（诸如，监视器）的显示数据。纹理映射用于将纹理施加至物件，图形处理系统 110 再现 3D 物件的纹理片，纹理片实质上经“堆叠”起来以形成物件。一旦建立了 3D 纹理，便将纹理施加给物件以形成影像，借此物件变得纹理化。

[0026] CPU 102 经由系统接口 108 向图形处理系统 110 提供请求，包括对处理并显示图形信息的请求。来自 CPU 102 的图形请求通过图形处理系统 110 接收，且提供给前端处理器 112。前端处理器 112 产生含有像素坐标的像素流。前端处理器 112 所产生的像素坐标的相关信息经由纹理快取系统 114 提供至纹理过滤器 118。纹理快取系统 114 自前端处理器 112 接收信息且组织纹理数据，从而将数据储存在快取存储器中。

[0027] 纹理过滤器 118 随后通过执行（例如）双线性过滤、三线性过滤或其组合来过滤信息，且产生每一像素的纹理数据。除已知纹理过滤器组件（诸如线性内插器以及累加器）之外，纹理过滤器 118 亦包括可编程表过滤器 (table filter) 以用于结合其他纹理过滤器组件来提供特定过滤操作。纹理数据 106 为发送至帧缓冲器 120 的最终色彩数据的分量，其用于在显示元件 130 上产生显示。

[0028] 纹理快取系统 114 可包括多个快取存储器, 包括(例如)1 级(L1)快取存储器以及 2 级(L2)快取存储器。纹理信息是以分别的纹理元素而储存, 纹理元素在图形处理期间用于界定以像素坐标显示的色彩数据。纹理数据 106 自系统存储器 104 流至纹理快取系统 114, 且随后流至纹理过滤器 118。

[0029] 现参看图 2, 其说明图形处理系统内的纹理快取系统的某些基本组件。应注意出于简要的目的, 省略了对于(由本领域技术人员)理解纹理数据管理单元非必要的某些组件。图形处理系统 110 包括接收纹理数据 106 的纹理快取系统 114。数据在经转发至快取存储器以用于储存之前受到处理。对于某些实施例, 纹理快取系统 114 包括存储库搅和模块 210 以及 L2 快取存储器 220。

[0030] 存储库搅和模块 210 获取自外部存储器接收的纹理数据 106, 且在将纹理数据 106 转发至 L2 快取存储器 220 以作储存之前对数据执行存储库搅和操作。通常, 存储库搅和操作是指重新映射存储库, 以使储存在具有相同索引的两个不同存储库中的纹理数据可同时得以撷取, 而不用等到下一周期。

[0031] L2 快取存储器 220 耦接至存储库搅和模块 210, 且在存储库搅和模块 210 已处理自外部存储器接收的数据后储存纹理数据 106 以用于未来撷取。参看图 3, 其为一实施例的 L2 快取存储器组织格式的方块图。如以上所论述的, 纹理快取系统 114 可包括多个快取存储器, 包括(例如)L1 快取存储器以及 L2 快取存储器, L2 快取存储器 310 可包括多个快取线 320。每一条快取线 320 可进一步分为奇数区块 330 以及偶数区块 340, 偶数区块 340 用于处理来自偶数管线的纹理数据, 而奇数区块 330 用于处理来自奇数管线的纹理数据。在奇数区块 330、偶数区块 340 的每一个内, 可将 L2 快取存储器进一步分为数据的存储库 350。尽管所揭示每一区块包括八个存储库(存储库 0 至存储库 7), 但本领域技术人员应了解快取存储器可分为不同数目的存储库。通常, 存储库为纹理数据得以提取的手段之一。应注意 L2 快取存储器 310 可含有多个 L2 快取线 320, 包括(例如)64 个、128 个、256 个, 或 512 个快取线。此外, 每一 L2 快取线 320 可具有变化的宽度, 包括(例如)128 个、256 个、512 个、1024 个或 2048 个位的宽度。

[0032] 现参看图 4, 其绘示经由存储库搅和模块自外部存储器转送纹理数据至 L2 快取存储器的一实施例。储存在外部存储器 410 中的纹理数据 412 在其最终储存在 L2 快取存储器 418 中以用于未来撷取之前, 会先转发至存储库搅和模块 414。存储库搅和模块 414 在逐存储库的基础上处理并重组自外部存储器 410 接收的纹理数据 412, 且将数据转发至 L2 快取存储器 418。如图 4 中所展示, L2 快取存储器 418 可经再分为多个快取线 416, 快取线 416 可进一步再分为多个存储库 420。对于某些实施例, 每一快取线 416 的大小为 1024 位, 或 1Kb。此外, 每一快取线 416 可进一步再分为各自大小为 128 位的八个存储库 420。在准备储存在快取存储器 418 中时, 存储库搅和模块 414 将自外部存储器 410 接收的纹理数据 412 组织为独立的存储库以及快取线。一旦执行了存储库搅和操作, 便将纹理数据 412 转发至 L2 快取存储器 418 以用于储存。

[0033] 现参看图 5, 其说明存储库搅和操作的一实施例。在进行存储库搅和操作之前, 将自外部存储器接收的纹理数据分为快取线(例如, 快取线 0、快取线 1), 再将每一快取线进一步分为个别存储库 510。对于所示的实施例而言, 每一快取线包含八个存储库 510(存储库 0 至存储库 7)。应注意通常可以各种方法来撷取纹理数据, 包括在逐存储库的基础上。

对于在逐存储库的基础上撷取纹理数据的实施例而言,可在存储器存取期间撷取任何给定快取线内的任何存储库。于一实施例中,存储器存取可试图提取以下存储库:(快取线 0, 存储库 0)、(快取线 0, 存储库 2)、(快取线 1, 存储库 7) 以及 (快取线 1, 存储库 0)。对于此实例,因为试图存取具有相同索引的两个存储库((快取线 0, 存储库 0) 以及 (快取线 1, 存储库 0)), 所以出现存储库碰撞。结果,在此周期期间仅能撷取两个存储库中的一个,因而需要额外周期来撷取剩余存储库。通常,因为纹理数据本质上倾向于定域的,所以很可能会出现存储库碰撞,纹理数据很可能储存在彼此邻近储存的存储库中。

[0034] 图 5 中揭示存储库搅和操作的一实施例,其增加了同样编号存储库之间的距离,以便减少存储库碰撞的可能性。对于某些实施例,可在 L2 快取存储器 230 内执行存储库搅和操作。应注意因为在外部存储器中实施搅和操作可能会由于外部存储器为共用资源而需要改变关于存储器存取的所有模块,所以存储库搅和操作最好在 L2 快取存储器 230 内执行而不在外部存储器中执行,否则将最终增加硬件的整体复杂性。

[0035] 在执行存储库搅和操作时可遵循各种映射机制或搅和模式。对于某些实施例而言,存储库搅和操作可通过根据所使用的映射机制重新路由(route)L2 快取存储器 230 内的特定存储器位置(亦即,存储库)的指定纹理数据。对于某些实施例,利用以下映射机制或存储库搅和模式:

[0036]

存储库 0↔存储库 4

[0037]

存储库 1↔存储库 5

[0038]

存储库 2↔存储库 6

[0039]

存储库 3↔存储库 7

[0040] 亦即,将待储存在 L2 快取存储器 230 内的存储库 0 与存储库 4 中的数据重新映射且调换,以及将存储库 1 中的数据与存储库 5 中的数据调换等。以上搅和模式在图 5 中加以说明。应注意对于各种实施例而言,每隔一快取线即需执行存储库搅和操作以便达成具有相同索引数字的存储库之间的最大距离。

[0041] 通常,利用存储库搅和操作可用以达成 u 方向与 v 方向上的最佳距离。详言之,重点在于两个方向(u, v)上的距离在执行存储库搅和操作之后为相等量值。参看图 6,其说明具有 32BPE 分辨率的纹理数据的 4×8 占据面积(footprint)。每一存储库内的像素的数目为 128 位 /32BPE, 或每存储库 4 纹理元素。如图 6 中所见,将邻近快取线(例如,快取线 0 以及快取线 1)中同样编号存储库(例如,存储库 0)之间的距离指定为距离 A 且等于 4 像素。一旦执行存储库搅和操作,存储库之间的距离便增加,借此减少由于纹理数据的定域性而发生存储库碰撞的可能性。现参看图 7,其说明存储库搅和操作之后邻近快取线中的同样编号存储库之间的距离上的增加。为达到说明的目的,仅存储库 0 以及存储库 4 得以重新映射。一旦执行存储库搅和操作,存储库之间的距离便增加至新距离 A。现通过以下表达式来简单地计算新距离 A 的值:

[0042] 新距离  $A = \sqrt{(4^2 + 4^2)} = 5.657$

[0043] 因此,位于邻近快取线中具有相同索引的存储库之间的距离为相隔 5.657 像素。简要参看图 8,其说明存储库搅和操作之前以及之后占据面积的大小。占据面积 810 反映跨越四个快取线的具有 32BPE 解析度的纹理数据的  $8 \times 4$  占据面积。为了解两个存储库之间的距离上的增加,在此仅检查两个快取线。占据面积 820 显示来自占据面积 810 的两个快取线。因此,占据面积 820 的大小为  $4 \times 4$  存储库。同样编号存储库之间的距离为 4 像素(距离 A),产生 16 像素<sup>2</sup>的面积。占据面积 830 反映经由存储库搅和操作所达成的有效占据面积。占据面积 830 的面积为 (新距离 A)<sup>2</sup>,或  $5.657^2$ 。因此,归因于存储库搅和操作的面积上的百分比增加为:

[0044]

$$\frac{([新距离 A]^2 - 4^2)}{4^2} = 100\%$$

[0045] 如以上所见,归因于存储库搅和操作的面积上的有效增加为原始面积的百分之百。

[0046] 参看图 9,其绘示用于执行存储库搅和操作以减少存储库碰撞的方法的一实施例的流程图。在以下步骤中,可透过执行存储库搅和操作以增加邻近快取线中的同样编号存储库之间的距离,以便减少图形处理单元的 L2 快取存储器内的存储库碰撞的可能性。方法开始于步骤 910,自外部存储器接收纹理数据。接着在步骤 920 中,将纹理数据分为多个存储库以准备储存在快取存储器中。随后,通常将存储库转发至 L2 快取存储器以用于储存。然而,在步骤 930 中,首先对自外部存储器所撷取的纹理数据执行存储库搅和操作,再将纹理数据于转发至 L2 快取存储器以作储存之前,在逐存储库的基础上执行存储库搅和。在步骤 940 中,将经重新映射的纹理数据写入至 L2 快取存储器内的适当存储库。于一实施例中,返回参看图 5。(存储库 0,快取线 1) 中所含有的纹理数据得以重新映射且现被写入至 L2 快取存储器内的(快取线 1,存储库 4)。同样,(快取线 1,存储库 1) 中所含有的纹理数据得以重新映射且现被写入至 L2 快取区内的(快取线 1,存储库 5)。应强调对交替快取线执行步骤 930 中的存储库搅和操作,以便加大邻近快取线中同样编号存储库之间的距离。重复步骤 930 至 940 直至所有纹理数据得以处理并重新映射为止(判定步骤 950)。

[0047] 现参看图 10,其绘示用于执行存储库搅和操作以减少存储库碰撞的方法的替代实施例的流程图。对于此实施例而言,当在提取操作期间出现存储库碰撞时,可进行存储库搅和操作。自外部存储器接收纹理数据且随后在转发至 L2 快取存储器之前将其分为存储库。随后,当需要纹理数据时,自 L2 快取存储器提取数据(步骤 1030)。若侦测到存储库碰撞(判定步骤 1040 中的“是”条件),则对涉及存储库碰撞的存储库执行存储库搅和操作,以将涉及碰撞的存储库中的至少一个重新映射至另一存储库(步骤 1050)。

[0048] 于一实施例中,假定在给定周期期间自 L2 快取存储器撷取纹理数据的以下存储库:(快取线 0,存储库 0)、(快取线 0,存储库 7)、(快取线 1,存储库 5),以及(快取线 1,存储库 0)。由于此撷取试图同时自快取线 0 与快取线 1 撷取存储库 0,所以出现存储库碰撞。因此,在当前周期期间仅能撷取存储库中的一个,且需要额外周期来撷取涉及碰撞的第二存储库。因为已出现存储库碰撞(判定步骤 1040 中的“是”条件),所以需对存储库 0 执行存储库搅和操作,以便减少另一存储库碰撞出现的可能性。根据先前所论述的存储库搅

和模式,将来自两个快取线中的一个(快取线0或快取线1)的存储库0重新映射至存储库4。因此,当需要相同纹理数据时,因为现在撷取以下存储库:(快取线0,存储库0)、(快取线0,存储库7)、(快取线1,存储库5)以及(快取线1,存储库4),所以避免了存储库碰撞。应注意由于不存在存储库碰撞,故不再需要额外周期来撷取纹理数据的存储库。

[0049] 在用于执行存储库搅和操作的方法的替代实施例中,可在存储库搅和操作之前执行分析,借此首先检查待提取的数据的存储库。通常,若侦测到存储库碰撞,则执行存储库搅和操作。然而,在新存储库碰撞将由于存储库搅和操作而出现的情况下,可对先前所描述的一个实施替代存储库搅和模式。

[0050] 于一实施例中,假定在给定周期期间自L2快取存储器撷取纹理数据的以下存储库:(快取线0,存储库0)、(快取线0,存储库4)、(快取线1,存储库5)以及(快取线1,存储库0)。因为出现存储库碰撞((快取线0,存储库0)与(快取线1,存储库0)之间),所以通常对存储库中的一个(例如,存储库0)执行存储库搅和操作,以便减少另一存储库碰撞出现的可能性。若利用先前所论述的存储库搅和模式,则来自两个快取线中的一个(快取线0或快取线1)的存储库0经重新映射至存储库4。因此,当需要相同纹理数据时,因为现在撷取以下存储库:(快取线0,存储库0)、(快取线0,存储库4)、(快取线1,存储库5)以及(快取线1,存储库4),所以避免了存储库碰撞。然而,请注意新存储库碰撞((快取线0,存储库4)与(快取线1,存储库4)之间)由于利用先前所描述的存储库搅和操作而产生。因此,在另一实施例中,可执行“预搅和”分析来确定另一个新的存储库碰撞是否会由于存储库搅和操作而产生。若将出现新存储库碰撞,则实施另一存储库搅和模式以避免随后又存储库碰撞。于一实施例中,可代替利用以下替代存储库搅和模式:

[0051]

存储库0↔存储库7

[0052]

存储库1↔存储库6

[0053]

存储库2↔存储库5

[0054]

存储库3↔存储库4

[0055] 返回参看以上实例,存储库因此将经重新映射至以下存储库:(快取线0,存储库0)、(快取线0,存储库4)、(快取线1、存储库5),以及(快取线1,存储库3)。结果,不出现随后的存储库碰撞。

[0056] 本发明的实施例可在硬件、软件、韧体或其组合中实施。在某些实施例中,可通过储存于存储器中且通过适合的指令执行系统所执行的软件或韧体中实施此方法以及系统。于一实施例中,若在硬件中实施,则可通过此项技术中所熟知的以下技术的任何个或组合来实施方法以及系统:一或多个离散逻辑电路,其具有逻辑门以用于在数据信号上实施逻辑功能;专用集成电路(ASIC),其具有适当的组合逻辑门;一或多个可编程门阵列(PGA);现场可编程门阵列(FPGA)等。

[0057] 如熟知本发明的技术者所能了解的,流程图中的任何处理描述或步骤应理解为表

示包括用于实施处理中的特定逻辑功能或步骤的一或多个可执行指令的模块、程序码的片段或部分，且替代实施例包括在本发明的实施例的范畴内，其中可视所包含的功能性而以不同于所展示或所论述的次序执行功能（包括实质上同时或逆序执行）。

[0058] 本文中包含用于实施逻辑功能的可执行指令的有序列表的方法以及系统可在任何计算机可读介质中具体化，计算机可读介质可由指令执行系统、装置或元件使用或结合指令执行系统、装置或元件使用，本文中的系统诸如基于计算机的系统、含处理器的系统，或可自指令执行系统、装置，或元件提取指令且执行指令的其他系统。在此文献的上下文中，“计算机可读介质”可为可含有、储存、通信、传播或传送由指令执行系统、装置或元件使用或结合指令执行系统、装置或元件使用的程序的任何构件。计算机可读介质可为（例如，但不限于）电子、磁性、光学、电磁、红外或半导体系统、装置、元件或传播介质。计算机可读介质的更特定实例（非详尽列表）将包括以下者：具有一或多个导线的电连接（电子式）、携带型计算机磁片（磁性式）、随机存取存储器（RAM）（电子式）、只读存储器（ROM）（电子式）、电可擦可编程只读存储器（EPROM 或快闪存储器）（电子式）、光学纤维（光学式），以及携带型紧密光盘只读存储器（CDROM）（光学式）。注意到因为程序可经由（例如）纸或其他介质的光学扫描而电子地俘获，随后经编译、解译或若需要以适合的方式另外处理，且随后储存在计算机存储器中，所以计算机可读介质甚至可为纸或另一适合的介质，在其上印有程序。此外，本揭露内容的范畴包括以在经硬件或软件组态的介质中具体化的逻辑模块来具体化本发明的实施例的功能性。

[0059] 此外，应强调以上所描述的实施例仅为可能实施例的实例。在不背离本揭露内容的原则的情况下可对以上所描述的实施例进行许多变化以及修改。所有此等修改以及变化在本文中意欲包括在本揭露内容的范畴内且通过权利要求书保护。

[0060] 举例而言，基于本文所提供的描述，本领域技术人员应了解实施例包括用于减少 L2 快取存储器内的存储库碰撞的方法，方法包含以下步骤：自用以储存用于图形处理单元内的纹理过滤的纹理数据的外部存储器读取纹理数据；将纹理数据分为多个存储库；对存储库执行存储库搅和操作；以及将数据的存储库写入至 L2 快取存储器。对于某些实施例而言，将纹理数据分为多个存储库包含将纹理数据组织成具有对应索引的连续存储库，其中存储库储存在至少一快取线中，且将存储库组织成快取线。对于其他实施例而言，快取线各自的大小为 1024 位且存储库的大小为 128 位，以致每快取线存在八个存储库。在某些实施例中，在逐存储库的基础上执行将纹理数据分为多个存储库且将数据的存储库写入至 L2 快取存储器的步骤。在某些实施例中，执行存储库搅和包含重新映射位于交替快取线中的存储库。在某些实施例中，执行存储库搅和更包含将位于邻近快取线中的具有相同索引的存储库以关于彼此成具有 45 度的数值的角而置放。在某些实施例中，重新映射包含以下步骤：将存储库 0 与存储库 4 调换；将存储库 1 与存储库 5 调换；将存储库 2 与存储库 6 调换；以及将存储库 3 与存储库 7 调换。

[0061] 另一实施例包括用于处理图形的计算机系统，计算机系统包含外部存储器，其用以储存用于图形处理单元内的纹理过滤的纹理数据；存储库搅和模块，其用以自外部存储器接收纹理数据，将纹理数据分为多个存储库，以及对纹理数据的存储库执行存储库搅和操作；以及 L2 快取存储器，其耦接至存储库搅和模块，其中 L2 快取存储器可用以储存经重新映射的纹理数据。对于某些实施例而言，将纹理数据分为多个存储库，其中存储库储存在

至少一快取线中。在某些实施例中，一或多个快取线各自的大小为 1024 位。在某些实施例中，存储库的大小为 128 位以致每快取线存在八个存储库。在某些实施例中，存储库搅和模块进一步可用以对交替快取线内的存储库执行存储库搅和操作。在某些实施例中，存储库搅和模块可用以仅在自 L2 快取存储器提取纹理数据期间出现存储库碰撞时执行存储库搅和操作。在某些实施例中，存储库搅和模块可用以在自外部存储器接收纹理数据之后执行存储库搅和操作。在某些实施例中，存储库搅和模块基于以下映射机制来执行存储库搅和操作：将存储库 0 与存储库 4 调换；将存储库 1 与存储库 5 调换；将存储库 2 与存储库 6 调换；以及将存储库 3 与存储库 7 调换。

[0062] 又一实施例包括用于减少 L2 快取存储器内的存储库碰撞的方法，方法包含以下步骤：自 L2 快取存储器提取纹理数据，其中纹理数据用于图形处理单元内的纹理过滤；确定在提取纹理数据时是否出现存储库碰撞，其中纹理数据被分为位于 L2 快取存储器内的至少一快取线中的多个存储库；以及在已出现存储库碰撞时对存储库执行存储库搅和操作。对于某些实施例而言，执行存储库搅和的步骤包含根据交替快取线重新映射涉及存储库碰撞的存储库。在某些实施例中，重新映射包含以下步骤：将存储库 0 与存储库 4 调换；将存储库 1 与存储库 5 调换；将存储库 2 与存储库 6 调换；以及将存储库 3 与存储库 7 调换。在某些实施例中，将纹理数据分为多个存储库，其中存储库储存在至少一快取线中，其中快取线各自的大小为 1024 位，且存储库的大小为 128 位以致每快取线存在八个存储库。在某些实施例中，在逐存储库的基础上执行将纹理数据分为多个存储库且将数据的存储库写入至 L2 快取区的步骤。

[0063] 以上所述仅为本发明较佳实施例，然其并非用以限定本发明的范围，任何熟悉本项技术的人员，在不脱离本发明的精神和范围内，可在此基础上做进一步的改进和变化，因此本发明的保护范围当以本申请的权利要求书所界定的范围为准。

[0064] 附图中符号的简单说明如下：

- [0065] 100 : 计算机系统
- [0066] 102 : CPU
- [0067] 104 : 系统存储器
- [0068] 106 : 纹理数据
- [0069] 108 : 系统接口
- [0070] 110 : 图形处理系统
- [0071] 112 : 前端处理器
- [0072] 114 : 纹理快取系统
- [0073] 118 : 纹理过滤器
- [0074] 120 : 帧缓冲器
- [0075] 130 : 显示元件
- [0076] 210 : 存储库搅和模块
- [0077] 220 : L2 快取存储器
- [0078] 230 : L2 快取存储器
- [0079] 310 : L2 快取存储器
- [0080] 320 : 快取线

- [0081] 330 :奇数区块
- [0082] 340 :偶数区块
- [0083] 350 :存储库
- [0084] 410 :外部存储器
- [0085] 412 :纹理数据
- [0086] 414 :存储库搅和模块
- [0087] 416 :快取线
- [0088] 418 :L2 快取存储器 / 快取存储器
- [0089] 420 :存储库
- [0090] 510 :存储库
- [0091] 810 :占据面积
- [0092] 820 :占据面积
- [0093] 830 :占据面积
- [0094] A :距离

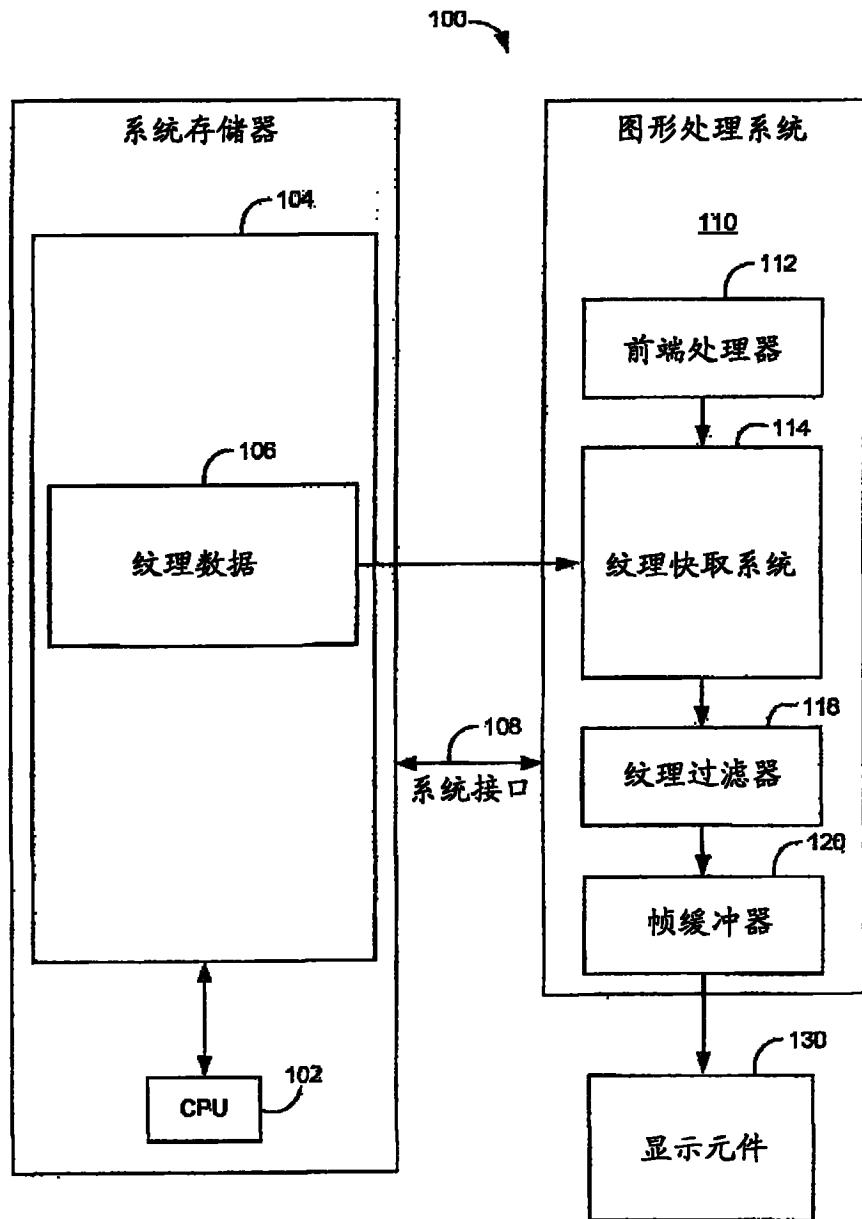


图 1

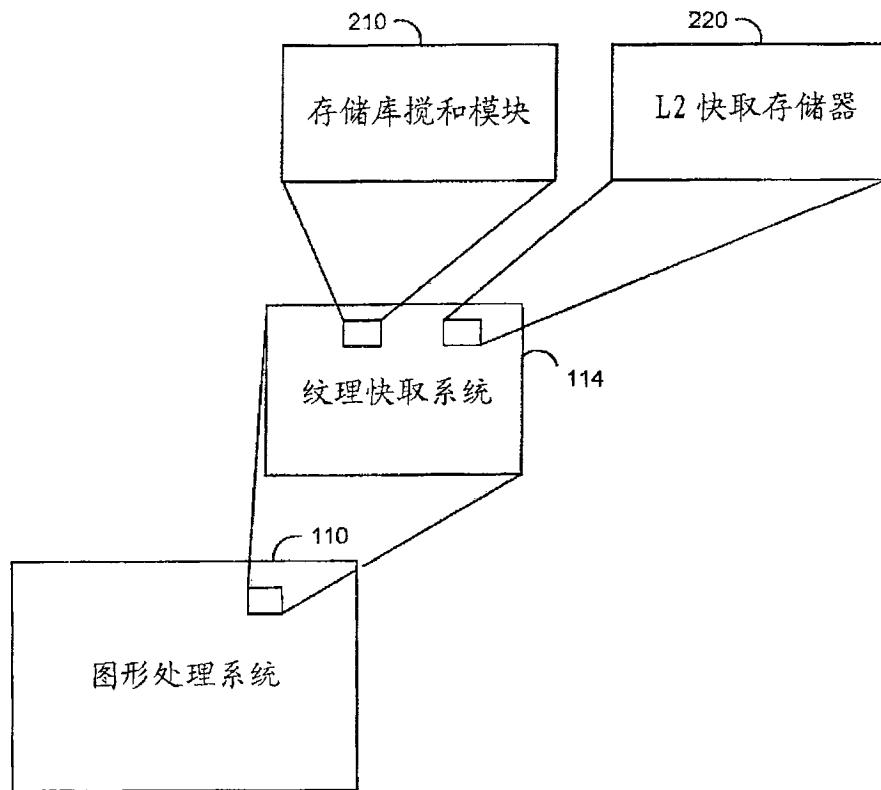


图 2

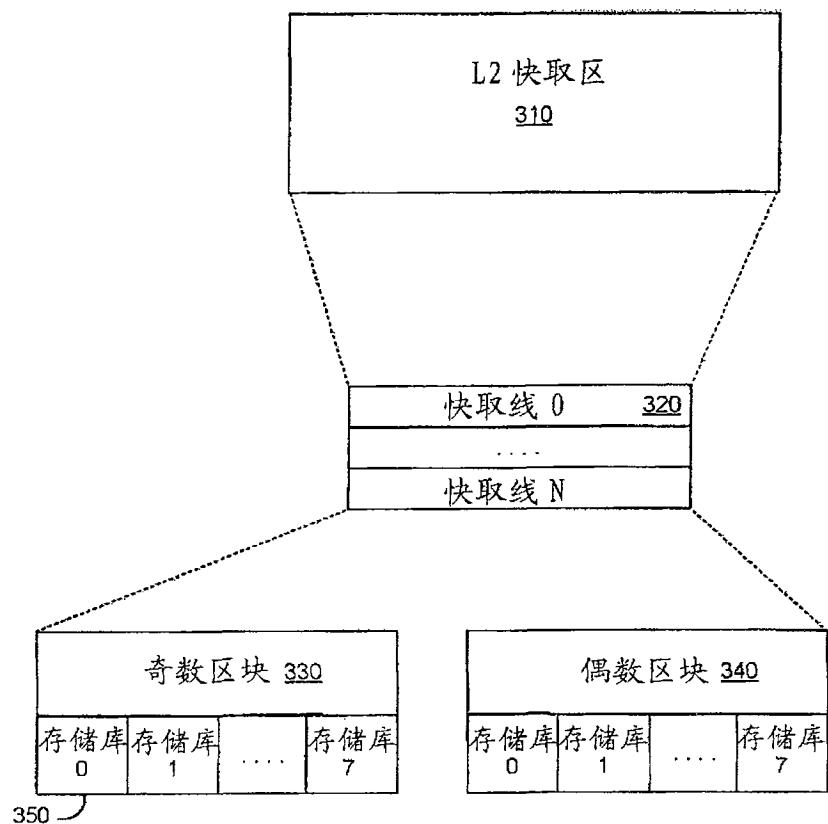


图 3

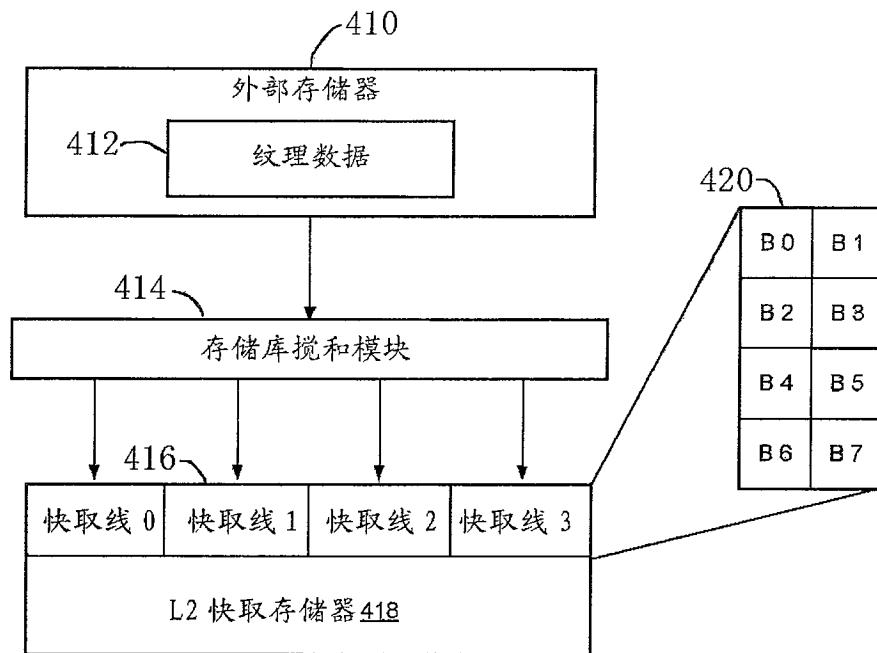


图 4

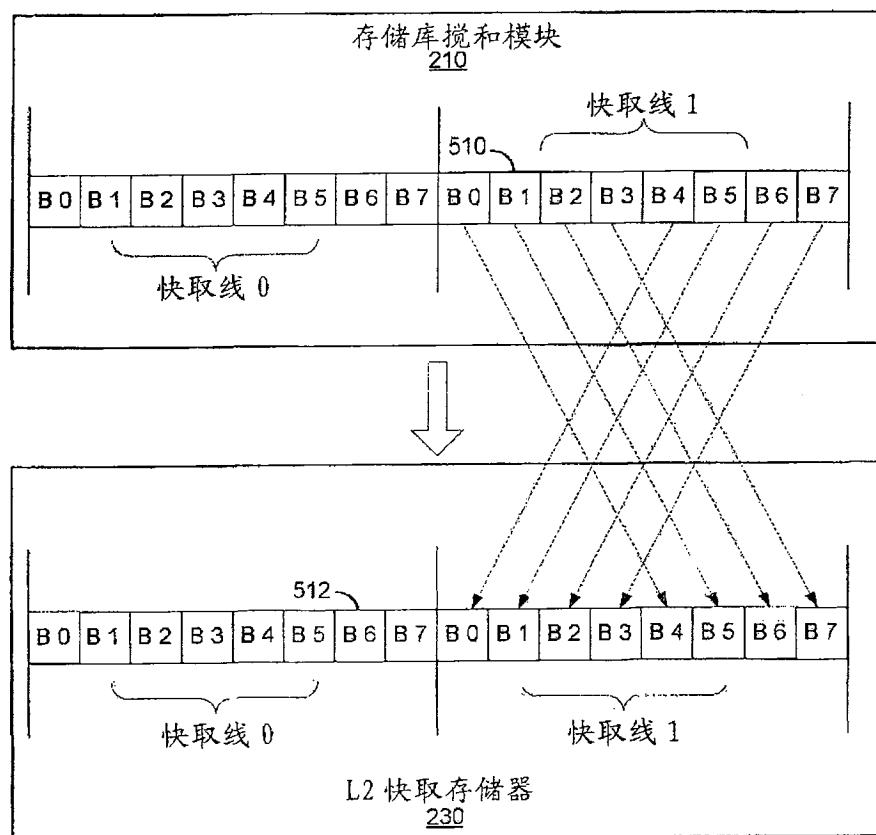


图 5

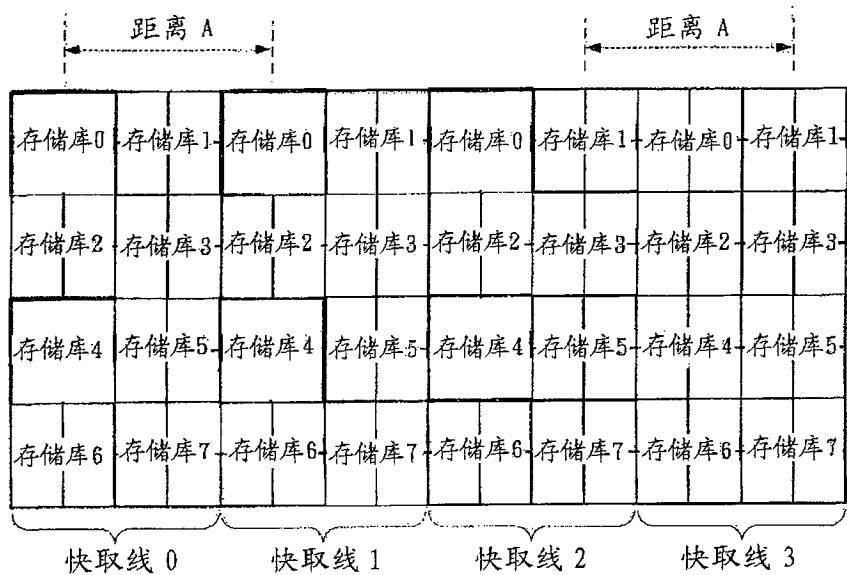


图 6

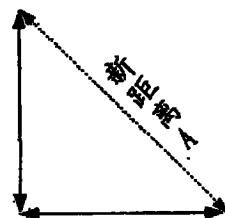
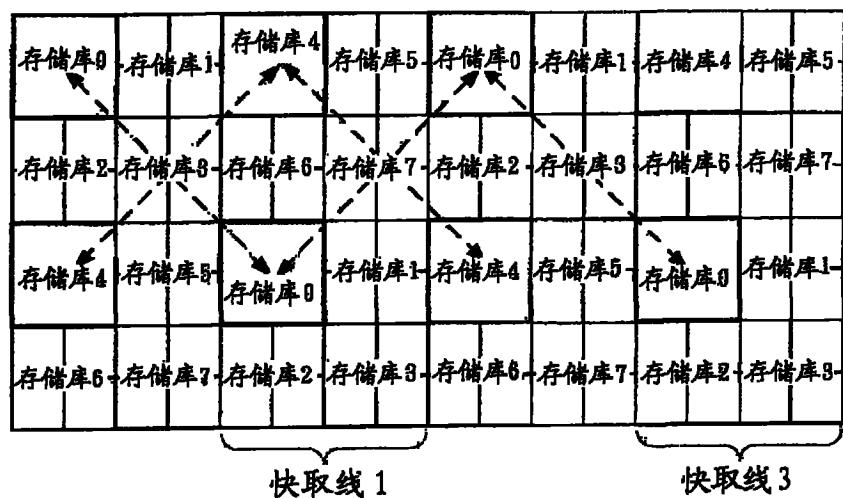


图 7

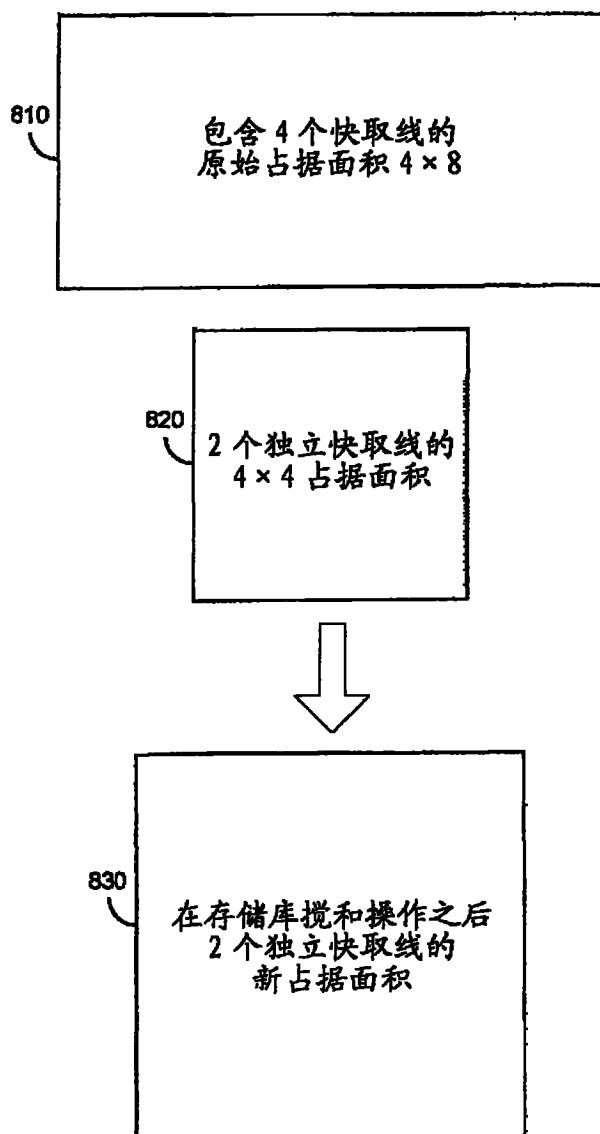


图 8

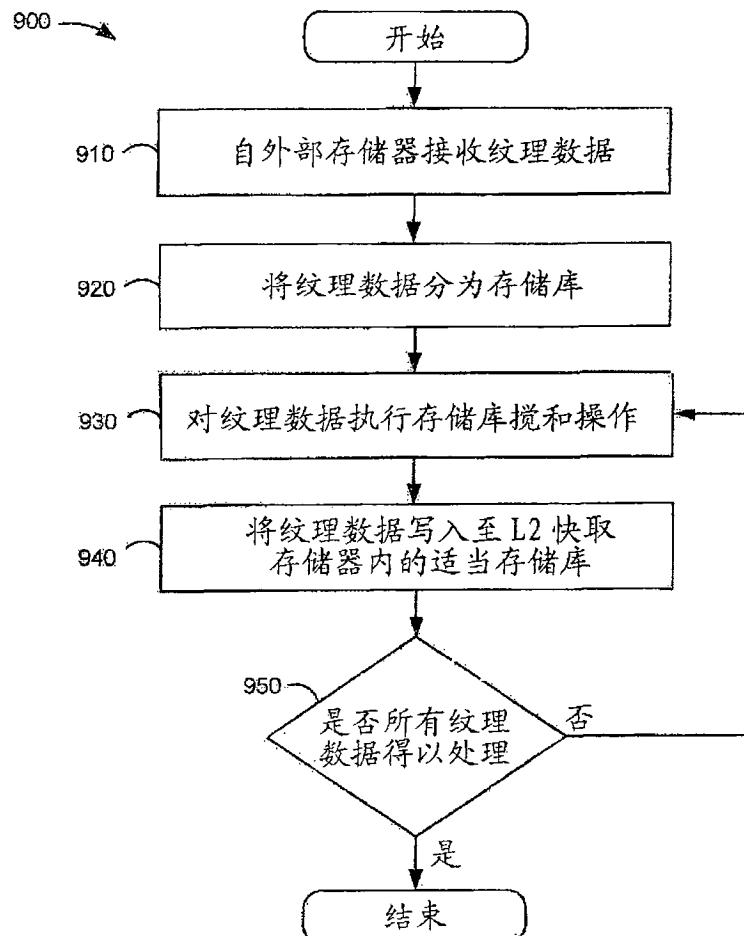


图 9

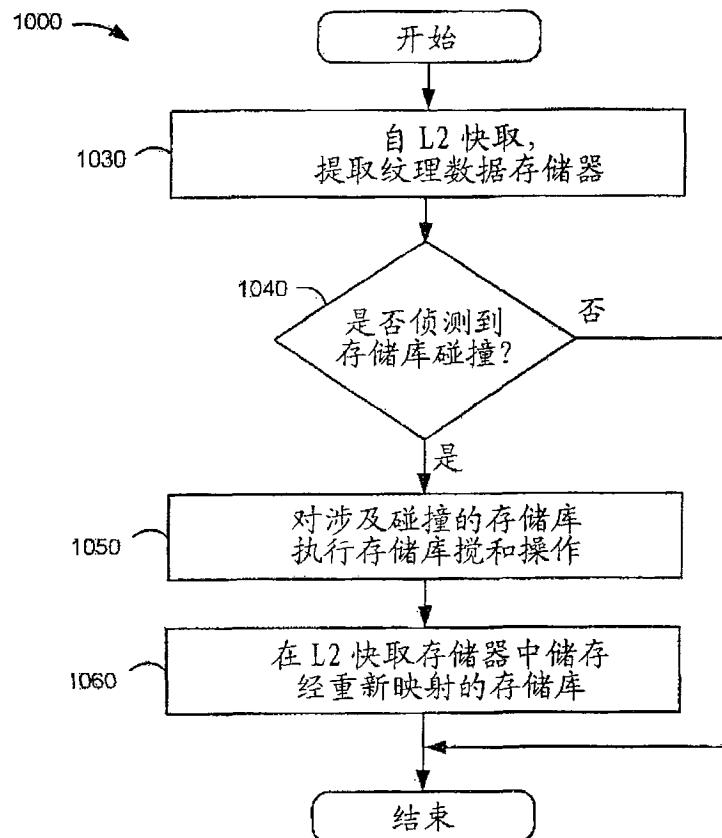


图 10