



(12) 发明专利

(10) 授权公告号 CN 103262425 B

(45) 授权公告日 2016. 05. 25

(21) 申请号 201180059927. 7

(51) Int. Cl.

(22) 申请日 2011. 12. 19

H03M 13/27(2006. 01)

(30) 优先权数据

61/424, 392 2010. 12. 17 US

13/329, 065 2011. 12. 16 US

(56) 对比文件

US 2005190736 A1, 2005. 09. 01,

US 7783936 B1, 2010. 08. 24,

US 2009199066 A1, 2009. 08. 06,

US 2008205636 A1, 2008. 08. 28,

US 2010077265 A1, 2010. 03. 25,

US 2008104478 A1, 2008. 05. 01,

US 2008091986 A1, 2008. 04. 17,

US 7814283 B1, 2010. 10. 12,

(85) PCT国际申请进入国家阶段日

2013. 06. 13

(86) PCT国际申请的申请数据

PCT/CN2011/084220 2011. 12. 19

(87) PCT国际申请的公布数据

W02012/079543 EN 2012. 06. 21

审查员 刘蕾蕾

(73) 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

(72) 发明人 王国辉 孙扬 约瑟夫·卡瓦拉罗

郭元斌

(74) 专利代理机构 北京同立钧成知识产权代理

有限公司 11205

代理人 张洋 黄健

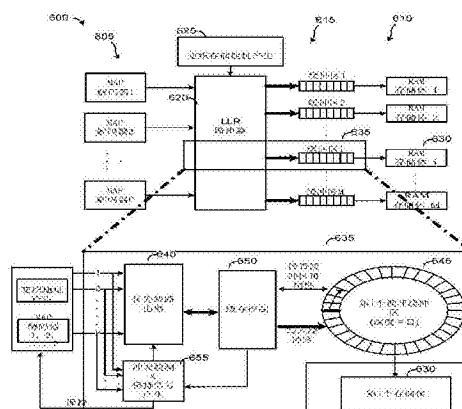
权利要求书3页 说明书15页 附图10页

(54) 发明名称

无竞争存储访问的系统和方法

(57) 摘要

一种 Turbo 码解码器的存储器控制单元包括一个具有多个存储槽的缓冲区, 一个缓存控制可操作地耦合到所述缓冲区, 一个路由器可操作地耦合到所述缓存控制和多个数据源, 一个冲突检测单元, 可操作地耦合到所述路由器、缓存控制、及多个数据源。该缓冲区临时存储用于存储在存储体中的信息。所述缓存控制确定在缓冲区中的可用存储槽数。所述路由器将数据从数据源发送到缓存控制。当可用存储槽的数目不足以存储所有试图访问内存块的数据源的数据, 所述冲突检测单元暂停一些数据源。



1. Turbo解码器的一个存储控制单元,其特征在于,包括:

第一缓冲区,具有多个存储插槽,所述第一缓冲区用于暂时存储在存储体中存储的信息;

缓存控制,其可操作地被耦合到所述第一缓冲区,所述缓存控制用于确定所述第一缓冲区中可用的存储插槽;

路由器,其可操作地被耦合到所述缓存控制和第二缓冲区,所述路由器用于将多个数据源的数据发送到所述缓存控制;

冲突检测单元,可操作地被耦合到所述路由器、所述缓存控制、所述第一缓冲区和所述第二缓冲区,所述冲突检测单元用于当可用存储槽的数目不足以存储所有试图访问存储体的数据源的数据时暂停一些数据源;

所述第二缓冲区,其可操作地被耦合到所述路由器、所述冲突检测单元、及所述多个数据源;所述第二缓冲区用于从试图访问存储体的数据源缓冲数据;其中,所述多个数据源为多个MAP解码器;

其中,所述第一缓冲区位于所述路由器和存储器之间;所述第二缓冲区位于所述多个数据源和所述路由器之间。

2. 根据权利要求1所述的存储控制单元,其特征在于,

所述冲突检测单元用于暂停操作N个数据源,其中N为整数,等于试图访问存储体的数据源的数量减去所述第一缓冲区中可用的存储槽数。

3. 根据权利要求1所述的存储控制单元,其特征在于,还包括:

旁路单元,其可操作地被耦合到所述缓存控制和存储体,所述旁路单元用于在所述第一缓冲区为空的情况下,从试图访问的存储体的多达L个数据源中存储数据,其中L是存储体中写端口的数量。

4. 根据权利要求1所述的存储控制单元,其特征在于,

所述存储控制单元与时钟周期序列同步工作;

在时钟周期序列的每个时钟周期中,所述缓存控制用于将所述第一缓冲区的多达L个存储插槽中存储的数据写入所述存储体中,其中L是存储体中写端口的数量。

5. 根据权利要求4所述的存储控制单元,其特征在于,L等于1。

6. 根据权利要求1所述的存储控制单元,其特征在于,

所述冲突检测单元用于在控制信号上设定一指定值,以临时暂停一些数据源。

7. 根据权利要求1所述的存储控制单元,其特征在于,

所述第二缓冲区用于将数据从多达K个试图访问存储体的数据源发送到所述路由器,其中K是同时允许的存储访问的最大数目。

8. 根据权利要求7所述的存储控制单元,其特征在于,

所述第二缓冲区用于从所述试图访问存储体的数据源的数据源中阻塞数据,其中所述试图访问存储体的数据源的数据源为试图访问存储体的数据源数量减去K个数据源。

9. 根据权利要求7所述的存储控制单元,其特征在于,

所述第二缓冲区是一个先入先出缓冲区。

10. 一个信息解码器,其特征在于,包括:

解码器,用于根据解码算法,对信号共同进行解码,并由此生成数据;

地址发生器,用于为所述解码器产生的数据生成存储器地址;

存储体,用于根据所述地址发生器生成的存储器地址,存储由所述解码器生成的数据;

存储控制单元,每个存储控制单元可操作地被耦合到所述解码器、其中一个存储体及所述地址发生器,具有多个存储插槽的每个存储控制单元用于暂时存储试图存储在其中一个存储体中的信息,且所述每个存储控制单元用于确定多个可用存储槽,将数据从所述解码器路由到存储槽,以及当可用存储槽的数目不足以存储来自于所述解码器中的试图访问其中一个存储体所有数据时,暂时阻塞一些解码器的运行;

其中,所述每个存储控制单元还用于缓冲来自于所述解码器的尝试访问其中一个存储体的数据。

11.根据权利要求10所述的信息解码器,其特征在于,

每个所述解码器都是一个软输入软输出解码器。

12.根据权利要求11所述的信息解码器,其特征在于,

每个所述解码器用最大后验概率算法解码信号。

13.根据权利要求10所述的信息解码器,其特征在于,

所述信号是一个turbo码解码的信号。

14.根据权利要求10所述的信息解码器,其特征在于,

所述每个存储控制单元用于,将尝试访问其中一个存储体的数据从多达K个解码器发送到路由器,其中K是在一个单个存储控制单元中同时允许的存储访问的最大数目。

15.根据权利要求14所述的信息解码器,其特征在于,

所述每个存储控制单元用于阻塞来自于多个解码器中的数据,所述多个解码器为试图访问其中一个存储体的多个解码器减去K个解码器。

16.一种操作一个存储控制单元的方法,其特征在于,所述方法包括:

从数据源接收数据,所述数据与访问存储体的整数N个并发存储关联;

确定缓冲区中的整数M个可用存储槽数;

当M大于或等于N时,暂时存储与访问所述缓冲区的N个并发存储关联的数据;

当N大于M时,暂时存储与访问所述缓冲区的N个并发存储中的M个并发存储关联的数据;

如果N大于M,停止N个并发存储访问中的N-M个访问;

所述方法还包括:

缓冲与N个并发存储访问相关联的数据。

17.根据权利要求16所述的方法,其特征在于,

所述存储控制单元与时钟周期序列同步工作,且所述方法还包括:

当所述缓冲区暂时存储至少一个数据片时,将与N个并发存储访问中多达L个相关联的数据写入,其中L是存储体中写端口的数量。

18.根据权利要求16所述的方法,其特征在于,所述停止包括:

在一个控制信号上设定一个指定值。

19.根据权利要求18所述的方法,其特征在于,缓冲数据包括:

将与N个并发存储访问的K个相关联的数据发送到所述缓冲区,其中K是同时允许的存储访问的最大数目;

如果 N 大于 K ,停止 N 个并发存储访问的 $N-K$ 个访问。

无竞争存储访问的系统和方法

[0001] 本申请要求于2010年12月17日提交美国专利局、临时申请号为61/424,392、发明名称为交织器中无竞争存储访问的系统和方法,于2011年12月16日提交美国专利局、非临时申请号为13/329065、发明名称为无竞争存储访问的系统和方法,其全部内容通过引用包含于本申请中。

[0002] 其内容通过引用包含于本申请中。

技术领域

[0003] 本发明一般地涉及信号处理,更具体地说,涉及一种无竞争存储访问的系统和方法。

背景技术

[0004] Turbo码是一类前向纠错(FEC)码,非常接近通信信道的信道容量(通信信道的码速率的理论最大值,此时在具体的噪声电平的情况下,有可能进行可靠的通信)。Turbo解码器是Turbo码的解码器。一般来说,一个Turbo解码器包括两个软输入软输出(SISO)解码器。SISO解码器可以利用一个最大后验概率(MAP)算法或一个软输出维特比算法(SOVA)对编码信号进行解码。

[0005] SISO解码器执行的算法,如MAP算法,SOVA等,往往是计算密集型的,因此,已经设计出了许多技术,以提高解码性能。一种常用的改善解码性能的技术是使用并行SISO解码,其中,使用所述多个SISO解码器,利用数据并行来提高解码性能。

发明内容

[0006] 本发明的实施例提供了一种无竞争存储访问的系统和方法。

[0007] 根据本发明的一个示例性实施例,提供了一种Turbo码解码器的存储器控制单元。所述存储器控制单元包括一个具有多个存储槽的缓冲区,一个缓存控制可操作地耦合到所述缓冲区,一个路由器可操作地被耦合到所述缓存控制和多个数据源,一个冲突检测单元,可操作地被耦合到所述路由器、缓存控制、及多个数据源。该缓冲区临时存储用于存储在存储体中的信息。所述缓存控制确定在缓冲区中的多个可用存储槽。所述路由器将数据从数据源发送到缓存控制。当可用存储槽的数目不足以存储所有试图访问存储体的数据源的数据时,所述冲突检测单元暂停一些数据源。

[0008] 根据本发明的另一示例性实施例,提供一个信息解码器。所述信息解码器包括解码器、一个地址发生器、存储体及存储器控制单元,其中每个存储器控制单元被可操作地耦合到解码器、到一个存储体及一个地址发生器。解码器根据解码算法共同对一个信号进行解码,并由此生成数据。地址发生器为由解码器产生的数据产生存储地址。根据由地址发生器产生的存储地址,所述存储体存储由解码器产生的数据。当可用存储槽的数目是不足以存储所有试图访问其中一个存储体的数据源的数据时,每一个存储器控制单元暂时存储用于存储在一个存储体中的信息,每一个存储器控制单元确定许多可用存储槽,将数据从解

码器发送到存储槽,并暂时阻塞一些解码器的运行。

[0009] 根据本发明的另一个示例性实施例,一种用于操作存储器控制单元的方法被提供。该方法包括:从数据源接收数据,该数据与对一个存储体的整数N个并发存储访问相关,并确定在一个缓冲区中的可用存储槽数,整数M。该方法还包括,如果M大于或等于N时,暂时存储与对缓冲区的N个并发存储访问相关的数据;当N大于M时,暂时存储与对缓冲区的N个并发存储访问中的M个访问相关的数据。所述方法还包括,如果N大于M,停止N个并发存储访问中的N-M个访问。

[0010] 一个实施例的一个优点是,为并行的SISO解码器提供了无竞争的存储访问,这样可以提高turbo解码器的性能。

[0011] 一个实施例的另一个优点是,所述硬件和软件的复杂性并未得到显著增加,这会对实施例的部署产生负面影响。

[0012] 一个实施例的另一个优点是,支持多个并行的SISO解码器的结构和相关算法。

附图说明

[0013] 为了更完整地理解本发明及其优点,现在参考以下的说明并结合附图,

[0014] 其中:

[0015] 图1表示根据上述实施例的示例性通信系统;

[0016] 图2表示根据上述实施例的turbo解码器的一部分的详细视图;

[0017] 图3a表示根据上述实施例的两个MAP解码器、一个数据存储器和一个交织器之间示例性关系图;

[0018] 图3b表示根据上述实施例的上半个迭代中的一个存储访问的示例图;

[0019] 图3c表示根据上述实施例的导致内存访问冲突的下半个迭代中的存储访问的示例图;

[0020] 图4表示根据上述实施例的turbo解码器示例图;

[0021] 图5a表示根据上述实施例的MAP解码器的不同程度并行性的内存冲突率与码块大小关系的示例图;

[0022] 图5b表示根据上述实施例的存储访问率与对单个存储体的存储访问次数关系的示例图;

[0023] 图6表示根据上述实施例的使用无竞争交织器的turbo解码器示例图;

[0024] 图7表示根据上述实施例的一个turbo解码器的一部分的示例图,其中详细描述一个带有旁路缓存的缓存控制单元;

[0025] 图8a和8b表示根据上述实施例,用基数-4PDFN MAP解码器生成数据的示例图;

[0026] 图9表示根据上述实施例的一个turbo解码器的示例图,其中所述turbo解码器包括双缓冲,以减少存储访问冲突;

[0027] 图10表示根据上述实施例的一个缓存控制单元详细视图;

[0028] 图11表示根据上述实施例,在缓存控制单元提供一个存储体的读取和写入访问时,缓存控制单元中进行的操作的流程图示例。

具体实施方式

[0029] 下面详细讨论了当前示例性实施例及其结构的操作。然而,应该理解,本发明提供了许多适用的发明概念,可以体现在多种特定的上下文中。所述特定实施例仅说明性地讨论了本发明的具体结构和操作本发明的方法,并不限制本发明的范围。

[0030] 本发明的一个实施例涉及防止存储访问中的竞争。例如,第一缓冲区用于缓冲向存储体进行的内存写入,以使内存写入能以正常的速度进行,而无需阻塞(或停止)内存写入源。第二缓冲区用来存储指定数量的同时进行的内存写入,从而总是允许指定数量的同步内存写入进行。

[0031] 本发明将对特定上下文中的优选实施例进行描述,即在turbo解码器中实现并行MAP解码器的通信设备。本发明适用于执行不同类型的并行SISO解码器和相关的算法的通信设备,例如MAP算法和SOVA。本发明可被应用于在各种通信网络中运行的通信设备,如在UMTS,3GPP LTE,WiMAX等技术标准可适用的通信网络中。此外,本发明可以被应用到能在多模中工作的通信设备(如那些可同时在两个或两个以上通信网络中工作的通信设备)。

[0032] 图1表示通信系统100。通信系统100包括第一通信设备105和第二通信设备110。第一通信设备105和第二通信设备110可通过传输无线信息进行通信。第一通信设备105和第二通信设备110包括发射器(发射器115和发射器120)和接收器(接收器117和接收器122)。由发射器115从第一通信设备105进行的传输能够由接收器122在第二通信设备110接收。同样,由发射器120从第二通信设备110进行的传输能够由接收器117在第一通信装置105接收。发射器115和120可使用turbo码解码信息,接收器117和122可以使用Turbo解码器解码信息。

[0033] 第一通信设备105和第二通信设备110还可以包括一个处理器(处理器119和处理器124)在传输之前或接收之后处理信息。该处理器也可用于控制通信设备的操作,以及执行应用程序和/或用户界面。

[0034] 如前所述,使用Turbo码和SISO解码器对它们进行解码已经取得了接近信道容量的性能。特别是,并行的SISO解码器的使用有助于提高解码性能。

[0035] 然而,并行的SISO解码器的使用会导致内存冲突,其中,一个以上的SISO解码器(例如,一种算法的一个实例,如MAP算法,SOVA等)尝试向一个单个存储体进行读取或写入。当有内存冲突时,对存储体的访问需要被序列化,以帮助确保内存的一致性。序列化存储访问能够击败由使用并行的SISO解码器而产生的性能提升,并显著降低了吞吐量。此外,随着并行的增加,内存冲突的百分比越来越高。因此,内存冲突已经成为turbo解码器应用程序,如在UMTS、3GPP LTE、WiMAX和其它标准都适用的通信设备中并行的SISO解码的性能瓶颈。

[0036] 虽然对下面提出的示例性实施例的讨论集中于MAP算法和MAP解码器,这是SISO解码器及其相关算法的例子,所述示例性实施例可与各种不同的SISO解码器一起运行。因此,MAP算法和MAP解码器的讨论不应该被解释为局限于示例性实施例的范围或精神中的任意一种。

[0037] 图2表示turbo解码器200的一部分的详细视图。Turbo解码器200可被设计来执行P个并行MAP解码器215,其中P是正整数。Turbo解码器200包括一个输入缓冲区205,其由M个存储体组成,其中M是正整数。所述M个存储体可以被用于存储输入信息(例如,对数似然比(LLR))。第一转换器210,例如,交叉转换器,可以将输入缓冲区205的M个存储体中的任意一个耦合到P个MAP解码器215中的任意一个。P个MAP解码器215的输出可以由第二转换器220

耦合到输出缓冲区225中的M个存储体中的任意一个,例如,交叉转换器。由第一转换器210和第二转换器220执行的转换可以由交织器地址发生器230进行控制和配置。

[0038] 在并行的turbo解码过程中的每个时间步长,每个MAP解码器从存储器中读取新的数据(例如,输入缓冲区205),以及将结果写入到存储器(例如,输出缓冲区225)。由于多个MAP解码器并行运行,向同一个存储体同时进行的多个存储访问能够发生,并需要多端口存储器或多个存储体来实现高吞吐量。由于多端口存储器的硬件实现效率普遍不高,多个存储体可在此处使用。通常,在并行turbo解码算法中,每一个MAP解码器被分配一个存储器。

[0039] 这里描述了一个内存冲突的例子。为了简化对这个问题的讨论,假定最小的块大小 $K=40$ 。并假定两个MAP解码器并行运行(即,平行性 $P=2$),并使用两个独立的存储体($M=2$)。MAP解码的一个完整的迭代包括两个半个迭代。在上半个迭代后,数据被交错,以使原始数据块中的比特间的依赖关系被更改或删除。图3a表示两个MAP解码器,一个数据存储器和一个交织器之间的关系。

[0040] 由于置换存储器内部的数据的成本很高,并且在存储器中的一个数据块的读取和写入的时延显著降低了吞吐量,动态地移动数据通常不是优选的。并不实际置换存储器中的数据,将数据保持在其原位置,并在运行中生成交错的内存访问地址是可取的。这些动态生成的内存访问地址表示置换后的数据的原始的数据存储器中的相应地址。通过使用这些交错的内存地址,在下半个迭代期间,所述MAP解码器知道从哪里获取所需要的数据,并且能够正确地从原来的存储器得到交错的数据。

[0041] 假设 $K=40$,上半个功能模块 $[d_0, d_2, \dots, d_{19}]$ 被存储在存储体A中,下半个功能模块 $[d_{20}, d_2, \dots, d_{39}]$ 被存储在存储体B中。根据图3b中所描绘的,在上半个迭代中,MAP解码器A从存储体A读取和写入数据,而MAP解码器B从存储体B中读取和写入数据。因此,在上半个迭代中并没有任何内存冲突。

[0042] 在下半个迭代中,所述数据应该以交错的方式从所述存储器中读取。由于交织算法几乎是随机的,如图3c所示,它很可能是两个MAP解码器尝试访问相同的存储体。图3c所示的并发存储访问导致内存冲突事件。

[0043] 在这个例子中,解码器的平行性只有2,当其平行性进一步增加时,如4,8,16等,可以有两个以上的MAP解码器同时尝试访问相同的存储体。内存冲突就能够成为系统性能的瓶颈。

[0044] 图4表示一个turbo解码器400的示例。turbo解码器400包括被耦合到存储器410的多个MAP解码器405(图4表示整数P个MAP解码器),存储器410可以被划分成M个存储体。路由器415耦合多个MAP解码器405到存储器410。例如,路由器415可被一个交织器存储器地址生成单元420所配置,它生成地址,这些地址指定多个MAP解码器405到存储器410中的存储体之间的连接。路由器415也可以被称为一个交织器或转换器。

[0045] 图5a表示MAP解码器不同并行程度的内存冲突率与块大小的关系图。当MAP解码器的并行性较低时($P=2$),所述内存冲突率保持在50%左右,但是,当并行性增加(P 大于或等于4),所述内存冲突率增加到90%以上。很明显,对于高的并行性,内存冲突成为turbo解码器及系统的整体性能中的一个严重的瓶颈。

[0046] 图5b表示对一个单个存储体的存储访问率与存储访问数量的关系图。如图5b所示,大多数内存冲突的产生是由于对单个存储体的两个或三个存储访问。因此,考虑一个解

决方案处理由对单个存储体的四个或更多的存储访问所产生的内存访问冲突,会不符合成本效益。

[0047] 表1表示存储体的内存冲突率和存储体数量之间的关系。如表1所示,通过将存储器分割成更多存储体,可以减少内存冲突率。然而,只是增加存储体的数量只能以有限的比例减少内存冲突。此外,使用更多的存储体,增加了硬件成本。因此,只增加更多的存储体,解决不了内存冲突问题,当与自由竞争的交织结构结合时,它应该被使用。

[0048] 表1内存冲突率和存储体数量之间的关系

[0049]

存储体数量	16	32	64	128
内存冲突率	100%	100%	97%	76.88%

[0050] 根据一个示例性实施例,目标可以是,减少和/或隐藏内存冲突,以使路由器415的时延尽可能的小,否则一个路由器(例如,路由器415)成为turbo解码器的瓶颈,并降低了turbo解码器的性能及通信系统的整体性能。因此,内存冲突发生时,使用小的缓冲区来存储数据(临时),理顺随时间推移不同的存储体之中的存储访问流量是可取的。

[0051] 图6表示使用无竞争交织器的turbo解码器600的示例。例如,turbo解码器600包括多个MAP解码器605(例如,P个MAP解码器),和被划分为M个存储体的一个存储器610中。此外,例如,存储器610的M个存储体中的每个可具有位于所述存储体和多个MAP解码器605之间的缓冲区。总的来说,缓冲区可以被作为缓冲区615。如图6所示,缓冲区615中有M个缓冲区,存储器610中的M个存储体中每一个都对应一个缓冲区。多个MAP解码器605的MAP解码器可由路由器620被耦合到存储器610中的存储体,例如,它可能会由交织器存储器地址生成单元625配置。

[0052] 根据一个实施例,存储器610的每个存储体(例如,存储体1630)都有存储体缓存控制单元635,以帮助提供无竞争的性能。对存储体缓存控制单元的详细讨论(例如,存储体缓存控制单元)如下。虽然讨论的焦点集中在一个单个存储体缓存控制单元,每个存储体可以有一个存储体缓存控制单元,它们在配置和操作上可以是完全相同的。因此对单个存储体缓存控制单元的讨论不应该被解释为被限制于所述实施例的范围或精神中的任意一种。

[0053] 在一个实施例中,存储体缓存控制单元635可包括一个优先级路由器640,这可以是路由器620的一部分,也可以被用来将一个MAP解码器耦合到缓冲器645和第i个存储体630。例如,一个缓存控制650可基于缓冲区645的状态提供控制信号。例如,缓存控制650能够确定,如果缓冲区645是空的、部分完整的、或完整的(例如,一个缓冲区645的状态),并根据缓冲区645的状态产生控制信号。缓冲区650也可以被允许对缓冲区645进行写入。

[0054] 在一个实施例中,冲突检测单元655可以被用来产生控制信号,这些控制信号可被提供给优先级路由器640,一个MAP解码器被耦合到第i个存储体630。例如,冲突检测单元655可检测对第i个存储体630同时进行的存储访问。如果对第i个存储体630同时进行的存储访问多于所允许的访问次数,或如果缓冲区645已满,则冲突检测单元655可以产生控制信号来阻塞(或保持或停止)被耦合到第i个存储体630的一个或多个MAP解码器。另外,在另一实施例中,在类似的情况下,冲突检测单元655可以产生控制信号,被提供给优先级路由器640。在一般情况下,术语阻塞、保持、停止等可以被互换使用,以描述一个运行的暂时中断,例如一个MAP解码器的运行。

[0055] 一般来说,缓冲器645被设计,暂时存储LLR数据包,其包括LLR数据和LLR数据的目的地址。在一个实施例中,缓冲区645的深度一般不是很大,一些数据可以同时被写入。两个指针分别用来控制缓冲区写入和读出。例如,在每一个时间步长,如果缓冲区645不是空的,缓冲区645将LLR数据写入第i个存储体630。如果缓冲区645中没有可用插槽,缓存控制650直接通知冲突检测单元655产生控制信号,来为所有尝试访问第i个存储体630的MAP解码器进行阻塞(或保持或停止)。

[0056] 冲突检测单元655为每一个从MAP解码器输出的LLR数据检查目的地址。冲突检测单元655首先选择第i个存储体630范围内的目的地址,然后对所选择的地址的数目进行计数。作为一个例子,如果选的地址数目大于2,则内存冲突发生。最后,冲突检测单元655将冲突信息传递给优先级路由器640或MAP解码器,例如,以所述控制信号的形式。

[0057] 如前所述,最频繁的内存冲突由2个或3个尝试访问单个存储体的MAP解码器所造成。尽管在最坏的情况下,可能有3个MAP解码器同时访问单个存储体,这些情况下通常是罕见的,因此平均情况下它们不会显著地影响系统的性能。在一个实施例中,在turbo解码器600中,重点是在最常见的情况上,其中内存冲突由2个或3个存储访问引起。

[0058] 在大多数情况下,对同一个存储体有不到3并发存储访问,优先级路由器640允许存储访问直接通过缓冲区645并将这些数据写入缓冲区645。根据一个实施例,在一个内存冲突事件中,当有3个以上的并发存储访问(冲突检测单元655所指示的)时,优先级的路由器640选择了其中3个,并将其写入到缓冲区645。与此同时,冲突检测单元655向不被允许的存储访问的MAP解码器发送HOLD信号。根据另一个实施例,优先级路由器640向不被允许的存储访问的MAP解码器发送HOLD信号。

[0059] 作为一个例子,当MAP解码器从冲突检测单元655收到HOLD信号,它们会阻塞(或停止)一个时钟周期。优先级路由器640使用几种不同的方法,来决定哪些数据可以被写入到缓冲区645中,哪些应保持。这些方法(这是示例性实施例的范围之外的,这里将不再讨论)用于在内存冲突期间,为每个可能的存储访问设置不同的优先级,从而确保每一个MAP解码器有一个相对公平的机会来输出其数据。这样做,也能够最大限度地减少整个缓冲区系统时延。

[0060] 根据一个实施例,缓存控制650可以被用来管理在缓冲区645内的读取和写入操作。例如,在每一个时间步长,缓存控制650检查缓冲区645内的空插槽的可用性。缓存控制650向优先级路由器640告知缓冲区645的可用插槽数。优先路由器640将根据可用插槽的数目,可以接收多少新数据,并决定由MAP解码器生成的哪个数据可以被写入到缓冲区645。

[0061] 根据一个示例性实施例,对缓冲系统的设计可能有一些设计权衡,如图所示。例如,需要考虑缓冲区645的深度。为了减少硬件资源的成本,在不增加时延的前提下,缓冲区645的深度需要尽可能地小。然而,当缓冲区645的深度减小,那么有时缓冲区645中的空插槽的数量并不足够容纳下一个时钟周期中的数据。这会通过断言HOLD信号,导致一些MAP解码器的阻塞(或停止)。断言HOLD信号通常会增加时延。因此,缓冲区深度需要慎重考虑。

[0062] 根据一个示例性实施例,由存储体缓存控制单元635所支持的同时进行的存储访问的数量可以是一个设计选择,该选择基于一些因素进行,例如turbo解码器600的所需性能、硬件和/或软件的成本等。例如,2个、3个、4个或更多的同时进行的存储访问可能被存储体缓存控制单元635支持,但是,更大数量同时存储访问的支持,能为缓存控制650和冲突检

测单元655引入更多的复杂性。此外,缓冲区645需要更大。

[0063] 虽然显示为一个深度为D的循环缓冲区,缓冲器645可以作为任何形式或类型的缓冲区来实现。因此,对一个循环缓冲区的讨论不应该被解释为被限制于所述实施例的范围或精神中的任意一个。此外,缓冲区645的大小可取决于对所支持的第i个存储体630的若干同时存储访问。在一般情况下,缓存区645的大小随同时进行的存储访问的增加而增加。

[0064] 根据一个示例性实施例,有几个参数会影响路由器的时延。表2至表5表示具有关键参数值的turbo解码器600的性能。这些参数包括:

[0065] 块大小(K):一个码字的块大小;

[0066] P:MAP解码算法的并行性,这是由并行的LLR数据的输出数量进行测量的,例如,如果所有的MAP解码器在一个时钟周期产生16个LLR值,P就是16;

[0067] M:存储体的数目;

[0068] m:优先级路由器的输出数,这个数目表示同时被允许写入到循环缓冲区的LLR数据的最大数目;

[0069] D:循环缓冲区的深度;

[0070] 被阻塞的MAP解码器的数目:这个数字表示在一个码字的解码过程中已经被阻塞的MAP解码器的数目;

[0071] 缓冲区几乎是满的情况的数目:这表示缓冲区几乎是满的,当时不能存储新数据的次数;

[0072] 理想的时钟周期(C0):这表示当假设没有内存冲突时,将所有的LLR数据写入内存所需的时钟周期数;

[0073] 实际时钟周期(C1):所有的LLR数据存储在存储器中所需的时钟周期的实际数目,包括缓冲系统的时延;

[0074] 时延(C1-C0):缓冲系统的时延;如果不使用缓冲区系统,时延将是一个非常大的数值。时延是一个值,可能会影响系统的吞吐量。缓冲系统的目的是最大限度地减少这个数值。

[0075] 表2表示优先级路由器输出的数量增加时,时延首先减少。然而,当MUX的数量大于4时,进一步增加优先级路由器的输出数量于事无补。这一事实与先前所示的存储访问模式的结果相匹配。表2:MUX数量不同的turbo解码器600的性能。

[0076]

参数					结果				
块大小	P	存储体的数目(M)	优先级路由器的输出数(m)	缓冲区深度	阻塞产生者的数目	缓冲区几乎是满的	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)
5114	16	16	1	15	2063	0	320	481	161
5114	16	16	2	15	157	0	320	341	21
5114	16	16	3	15	43	5	320	336	16
5114	16	16	4	15	28	11	320	335	15

[0077] 表3表示当缓冲区的大小大于一个特定的值,进一步增加缓冲区大小并不能有助于减少时延,尤其是当缓冲区足够大时,缓冲区在解码过程中几乎是从来没有满的。

[0078] 表3:缓冲区大小不同的turbo解码器600的结果

[0079]

参数					结果				
块大小	P	存储体的数目(M)	优先级路由器的输出数(N_选择)	缓冲区深度	阻塞产生者的数目	缓冲区几乎是满的	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)

[0080]

5114	16	16	3	5	454	325	320	366	46
5114	16	16	3	10	101	66	320	336	16
5114	16	16	3	15	43	5	320	336	16
5114	16	16	3	20	52	0	320	335	15

[0081] 表4表示,增加存储体的数量(M)有助于显著地减少时延。然而,当M大于64,进一步增加M通常不会进一步显著减少时延。

[0082] 表4:存储体数目不同的turbo解码器600的结果

[0083]

参数					结果				
块大小	P	存储体的数目(M)	优先级路由器的输出数(N_选择)	缓冲区深度	阻塞产生者的数目	缓冲区几乎是满的	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)
5114	16	16	4	20	13	0	320	335	15
5114	16	32	4	20	3	0	320	329	9
5114	16	64	4	20	3	0	320	324	4
5114	16	128	4	20	1	0	320	324	4

[0084] 表5表示以缓冲区为基础的结构也可以很好地作用于其它的块大小。不同的块大小的时延值是非常小的,所以它们一般不会影响系统的吞吐量。

[0085] 表5:块大小不同的turbo解码器600的结果

[0086]

参数					结果				
块大小	P	存储体数量(M)	优先级路由器的输出数(N_选择)	缓冲区深度	阻塞产生者的数目	缓冲区几乎是满的	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)

[0087]

						满的			
1024	16	64	3	10	0	0	64	66	2
2048	16	64	3	10	0	0	128	129	1
3072	16	64	3	10	0	0	192	194	2
4096	16	64	3	10	0	0	256	258	2

[0088] 缓冲区的使用,如缓冲区645,可以解决存储访问冲突的问题,并显著减少由频繁的内存访问冲突引入的时延。然而,当一个存储体的缓冲区是空的,就没有必要将数据写入到缓冲区。相反,数据可以被直接写入到存储体。

[0089] 图7表示在一个示例性实施例中turbo解码器700的一个部分,其中,详细表示了具有旁路缓存735的缓存控制单元705。如图7所示,缓存控制单元705可被耦合到第i个存储体710、MAP解码器及一个交织器地址生成单元。缓存控制单元705包括一个优先级路由器715、一个缓冲区720(如图7所示的一个循环缓冲区,但也可以使用其它的缓冲区类型)、一个缓存控制725和一个冲突检测单元730。作为一个示例性实施例,优先级路由器715、缓冲区

720、缓存控制725和冲突检测单元730可与图6的缓存控制单元635的相应的电路类似。

[0090] 根据一个实施例,缓存控制单元705还包括被耦合到缓存控制725和第i个存储体710之间的旁路缓存735。当缓冲区720是一个空的状态,旁路缓存735可以允许缓存控制725直接将数据写入到第i个存储体710,而不是将数据写入缓冲区720,然后将缓冲区720的内容写入到第i个存储体710。

[0091] 旁路缓存735的使用允许将数据直接写入第i个存储体710,而不是将数据写入缓冲区720,然后在后续的存储器周期中,将数据从缓冲区720写入到第i个存储体710。通过使用旁路缓存735,一个存储周期可以得到保存。表6表示旁路缓存对turbo解码器700的整体性能的影响。表6表示旁路缓存的使用显著减少了时延,并进一步允许在缓冲区的大小的减少,而对性能不产生负面影响的情况。

[0092] 表1:旁路模式的结果

[0093]

参数	结果
----	----

[0094]

结 构	块 大 小	P	存储 体数 量 (M)	优先级 路由器的输出 数(N_选择)	缓 冲 区 深 度	阻塞产 生者的 数目	缓冲 区几 乎是 满的	理想的 时钟周 期(C0)	实际时 钟周期 (C1)	时延 (C1-C0)
无 旁 路	5114	16	64	3	7	10	3	320	324	4
旁 路	5114	16	64	3	7	10	0	320	322	2
旁 路	5114	16	64	3	3	10	7	320	322	2

[0095] 前述的实施例都集中在基数-2单流式MAP解码器。然而,对于更高的基数MAP解码器,例如,对一个基数-4PDFN的MAP解码器,一种阻塞、保持或停止技术不足以提供良好的结果。基数-4PDFN MAP解码器的使用仅供参考,所述实施例并不限于基数-4PDFN MAP解码器。该实施例可与任何高基数的MAP解码器和任何并行度不同的并行turbo解码器一起运行。因此,对基数-4PDFN MAP解码器的讨论不应该被解释为局限于所述实施例的范围或精神中的任意一种。

[0096] 图8a和8b表示由一个基数-4PDFNMAP解码器进行的数据生成。如图8a中所示,基数-4PDFNMAP解码器在每个时钟周期生成四个数据单位。因此,如果使用前面描述的能够解

决三个同时存储访问的那些类似的缓冲技术,那么尽管四个数据单元中其实只有一个导致内存冲突,每次产生四个数据单位时,基数-4PDFNMAP解码器被阻塞、保持或停止。基数-4PDFNMAP解码器的一再阻塞,会导致时延的显著增加。

[0097] 图9表示turbo解码器900,其中,所述turbo解码器900包括双缓冲区,以减少内存访问冲突。turbo解码器900包括多个MAP解码器910和路由器915之间的第一缓冲区905,及路由器915和存储器925之间的第二缓冲区920。第一缓冲区905可以包括多个缓冲区(例如,先入先出缓冲区),每个缓冲区被耦合至一个MAP解码器的输出。根据一个示例性实施例,如果一个MAP解码器在每个时钟周期生成一个以上的数据输出,足够数量的缓冲区可用来为MAP解码器的输出提供缓冲。多个MAP解码器910中的MAP解码器,可以直接将数据写入到第一个缓冲区905,路由器915将与第一缓冲器905,而不是与多个MAP解码器910交互。一个缓存控制单元930将多个MAP解码器910连接到第M个存储体935。

[0098] 图10表示一个缓存控制单元1005的详细视图1000。图10提供缓存控制单元1005的详细视图,它将第i个存储体1010连接到多个MAP解码器和一个交织器地址生成单元。缓存控制单元1005与图9所示的缓存控制单元930类似。

[0099] 在一个实施例中,缓存控制单元1005包括:优先级路由器1015、第一缓冲器1020(如图10中所示的一个循环缓冲区,但也可以使用其它类型的缓冲区)、缓存控制1025,冲突检测单元1030和旁路缓存1035。作为一个例子,优先级路由器1015、第一缓冲区1020、缓存控制1025、冲突检测单元1030及旁路缓存1035与图7的缓冲控制单元705中的相应的电路相类似。

[0100] 根据一个实施例,缓存控制单元1005还包括多个MAP解码器和优先级路由器1015之间耦合的第二缓冲区1040。第二缓冲器1040类似于图9所示的第一缓冲区905。优先级路由器1015能与第二缓冲器1040,而不是与多个MAP解码器交互。在每个时钟周期,第二缓冲区1040中的每个非空的缓冲区尝试输出一个数据,优先级路由器1015检查第二缓冲区1040的输出的存储访问模式。在必要时,冲突检测单元1030可以向第二缓存区1040断言一个保持信号,与一个导致内存访问冲突的MAP解码器相对应的一个缓冲区将保持输出数据,并再次在随后的时钟周期内尝试输出数据。根据另一种示例性实施例,冲突检测单元1030可以向一个优先级路由器1015断言一个保持信号,以保持数据的发送。

[0101] 一个高基数MAP解码器,如基数-4PDFN MAP解码器,与第二个缓冲区1040中的缓冲区一起,可以被视为多个独立的普通单流式MAP解码器。在第二个缓冲区1040中的缓冲区将MAP解码器的输出再次独立,使优先级路由器1015可独立地保持、阻塞、或停止它们。双缓冲区结构的示例性实施例具有以下几个优点:

[0102] -不需要阻塞或停止MAP解码器。相反,在必要时缓冲区中的数据被保持。因此,UMTS中MAP解码器的结构与LTE/WiMAX标准可以是相同的。一致的MAP解码器结构可以减少MAP解码器的复杂性,并使多模式的解码器的设计更容易;

[0103] -存储访问更加顺利,因此,进一步减少缓冲区深度是有可能的。

[0104] 表7至表10表示单缓冲区结构(如图7)与双缓冲区结构(如图10)之间的比较结果。表7表示不采用旁路缓存机制的两种结构的结果。如表7中的第二列和第三列所示,双缓冲区结构进一步降低了实现低时延所需的最小缓冲区的深度。此外,表8表示采用旁路缓存机制的两种结构的模拟结果。

[0105] 采用双缓冲区结构,由于MAP解码器并不被阻塞,所述缓冲区进一步使内存访问流量顺利进行,在不增加时延的情况下,所述缓冲区深度可以进一步(使用或不使用旁路缓存)减少。此外,该缓冲区的大小仅是3,因此将FIFO添加到缓冲系统中将不会显著增加硬件成本。

[0106] 请注意,并行性P表示PDFN MAP解码器的数量。每个PDFN MAP解码器在一个时钟周期产生4个LLR数据。因此,并行LLR数据输出的并行性等于4×P。

[0107] 图7:不采用旁路机制的单缓冲区结构和双缓冲区结构的比较结果

[0108]

参数							结果				
结构	块大小	P	存储体数量	N_选择	FIFO深度	缓冲区深度	阻塞产生者的数目	缓冲区几乎溢出	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)
原本的	5120	4	64	3	-	7	35	5	320	328	8
双缓冲	5120	4	64	3	3	7	9	0	320	323	3

[0109]

缓冲区											
双缓冲区	5120	4	64	3	3	5	10	5	320	323	3

[0110] 表8:采用旁路机制的单缓冲区结构和双缓冲区结构的比较结果,K=5120,P=4

[0111]

参数							结果				
结构	块大小	P	存储体数量	N_选择	FIFO深度	缓冲区深度	阻塞产生者的数目	缓冲区几乎溢出	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)
原本的	5120	4	64	3	-	3	31	26	320	324	4
双缓冲区	5120	4	64	3	3	3	9	0	320	322	2
双缓冲区	5120	4	64	3	3	2	9	9	320	322	2

[0112] 表9表示单缓冲区结构和双缓冲区结构之间的内存比较。比较表明,双缓冲结构使用更少的内存,可以达到相同的时延性能。

[0113] 表9:单缓冲区结构和双缓冲区结构之间的内存要求比较

[0114]

	缓冲区深度	缓冲区数量	整体的缓冲区大小	FIFO深度	FIFO数量	整体的FIFO大小	整体的存储位置
原本的结构	3	64	192	0	16	0	192
双缓冲区结构	2	64	128	3	16	48	176
差别	-	-	-64	-	-	48	-16

[0115] 表10表示不同块大小的结果。结果表示双缓冲区结构可作用于不同的块大小。

[0116] 表10:不同块大小的双缓冲区结构的结果。

[0117]

参数						结果				
块大小	P	存储体数量	N_选择	FIFO深度	缓冲区深度	被阻塞的FIFO的数目	缓冲区几乎溢出	理想的时钟周期(C0)	实际时钟周期(C1)	时延(C1-C0)
1200	4	64	3	3	2	5	5	75	77	2
2400	4	64	3	1	2	0	0	150	150	0
3600	4	64	3	3	2	9	9	225	227	2
5114	4	64	3	3	2	9	9	320	322	2

[0118] 从表10中所示的结果可知,当块大小为K=5114,平行性P=16,优先级路由器一次选择多达3个数据,FIFO深度为3,在缓冲深度D=3时,双缓冲区结构的时延是2个时钟周期。值得注意的是,缓冲区路由器中控制逻辑的时延被计为一个时钟周期。

[0119] 根据上述分析及实施结果,有可能得出结论:双缓冲区结构的UMTS交织器可以显著减少内存访问冲突造成的时延。特别是,加入旁路缓存后,时延可被进一步减少,而硬件成本不会显著增加。如上所示,利用此处所述的双缓冲区结构的turbo解码器实现了性能,其值接近一个使用理想交织器的一个模型turbo解码器的解码吞吐量,该交织器不具有内存冲突。

[0120] 上述讨论的重点是从MAP解码器向内存写入。在上半个迭代中,存储访问是按顺序读取的。每个MAP解码器只能从相应的内存读取数据。在下半个迭代中,数据从内存中以交错的方式读取,这会导致内存冲突问题。不过,可以通过在两个半个迭代中以交错的方式写入数据来处理内存读取冲突。由于已知每次迭代中内存读取地址模式,根据交织算法写入数据是有可能的,因此从所述内存中读取数据时,按顺序读取数据是有可能的。例如,MAP解码器i只能从存储体i读取。因此,通过从MAP解码器自身的存储体中按顺序读取和基于交织算法写入数据,可避免读取存储访问中的冲突。

[0121] 图11说明当缓存控制单元向一个存储体提供读取和写入访问时,发生在缓存控制单元中的操作1100的流程图。操作1100可以指示当所述缓存控制单元向耦合到缓存控制单元的MAP解码器提供存储访问时,发生在一个缓存控制单元,如缓存控制单元705或缓存控制单元1005中的操作。缓存控制单元在正常操作模式时,操作1100能够发生。

[0122] 操作1100可以开始于缓存控制单元,其确定向一个正在运行的存储体进行的(N)个并发存储访问(功能模块1105)。根据一个实施例,如果一个存储访问是向其存储体中的一个内存位置进行的,优先级路由器可以从存储器地址确定。然后,缓存控制单元可以将并发存储访问的数量(N)与所允许的最大数量的并发存储访问(功能模块1110)进行比较。

[0123] 如果并发存储访问的数目(N)小于或等于并发存储访问(K)的最大数目(功能模块1110),那么缓存控制单元可允许所有的N个并发存储访问进行(功能模块1115)。但是,如果并发存储访问的数目(N)大于并发存储访问(功能模块1110)的最大数目,那么FIFO中,所述

缓存控制单元会阻塞大于并发存储访问最大数目的那些并发存储访问(即, N -并发存储访问的最大数目, 或仅仅 $N-K$)(例如, 图10中的第2缓冲区1040)。

[0124] 根据一个实施例, 所述缓存控制单元能够阻塞并发存储访问, 该阻塞可通过在MAP解码器上断言一个保持信号来实现, 该MAP解码器产生将被阻塞的存储访问(适用于单缓冲区结构); 或者通过MAP解码器的缓冲实现, 所述MAP解码器产生将被阻塞的存储访问(适用于双缓冲区结构)。

[0125] 对于所允许进行的 N 个并发存储访问, 所述缓存控制单元可以进行检查, 以确定与存储体相关的缓冲区是否是空的(功能模块1125)。如果所述缓冲区是空的, 那么, 缓存控制单元可直接向存储体执行其中一个存储访问(例如, 一个内存写入), 而不是将与内存写入相关的数据直接写入缓冲区(功能模块1130)。缓存控制单元然后再递减 N (功能模块1135)。值得注意的是, 如果所述存储体是一个多重写端口存储体, 例如, 一个 L 端口存储体, 那么缓存控制单元可以向存储体执行 N 个存储访问(例如, 内存写入)中的 L 个访问。

[0126] 如果所述缓冲区不是空的, 或如果缓存控制单元已进行一次存储访问, 缓存控制单元可以执行检查, 以确定所述缓冲区能保持的数据量 M (功能模块1140)。根据一个实施例, 缓存控制单元确定缓冲区中的可用空间量。

[0127] 所述缓存控制单元然后可以执行检查, 以确定所述缓冲区能保持的数据量(M)是否大于或等于并发存储访问的数量(N)(功能模块1145)。如果 M 小于 N , 则所述缓冲区必须阻塞不能存储在缓冲区(功能模块1150)中的 $(N-M)$ 个并发存储访问。

[0128] 根据一个实施例, 所述缓存控制单元能够阻塞并发存储访问, 该阻塞可通过在MAP解码器上断言一个保持信号来实现, 该MAP解码器产生将被阻塞的存储访问(适用于单缓冲区结构); 或者通过MAP解码器的缓冲实现, 所述MAP解码器产生将被阻塞的存储访问(适用于双缓冲区结构)。

[0129] 如果 M 大于或等于 N , 则缓存控制单元可以设置 N 等于 M (功能模块1155), 并存储与对所述缓冲区(功能模块1160)的 N 个并发存储访问相关的数据。操作1100然后终止。

[0130] 尽管参照前述实施例对本发明进行了详细的说明, 本领域的普通技术人员应当理解: 其依然可以对前述各实施例所记载的技术方案进行修改, 或者对其中部分技术特征进行等同替换; 而这些修改或者替换, 并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

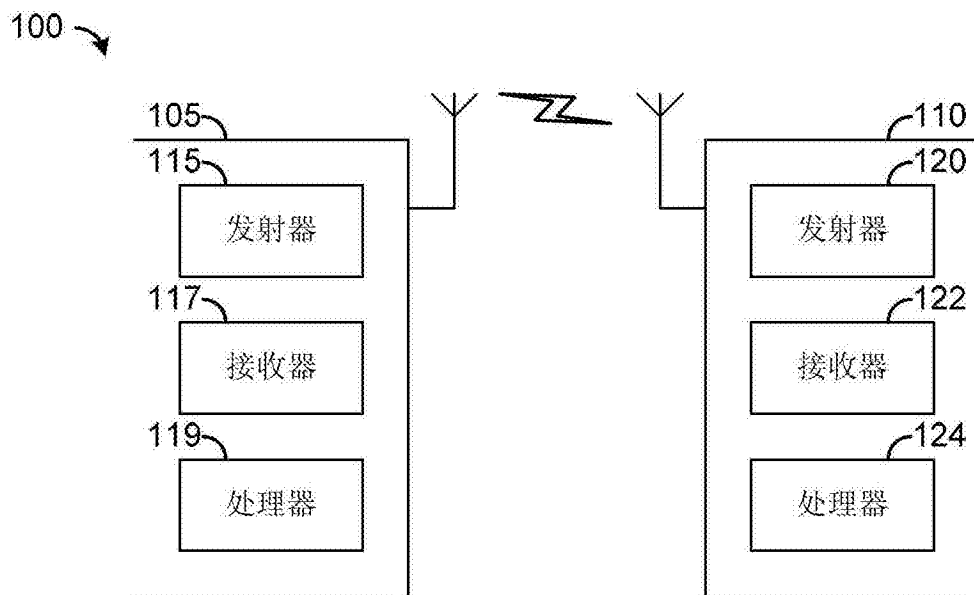


图1

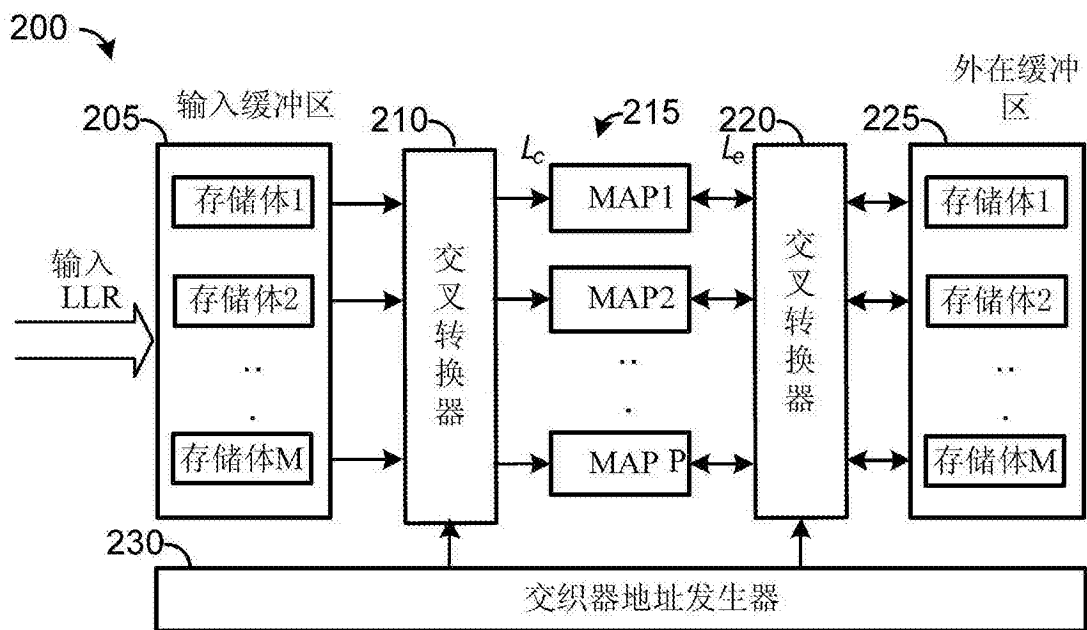


图2

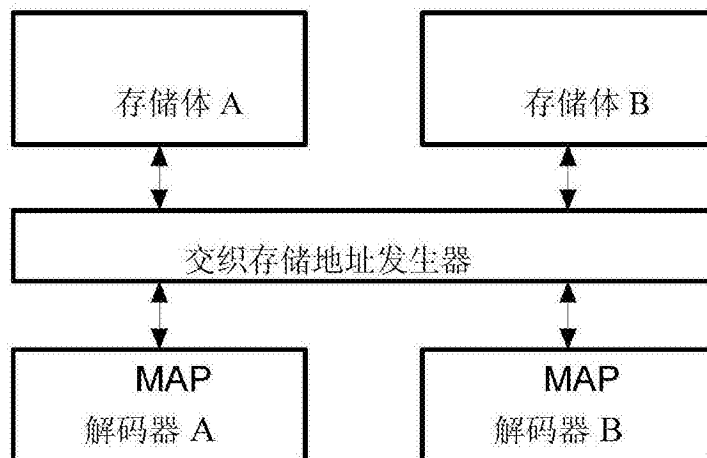


图3a

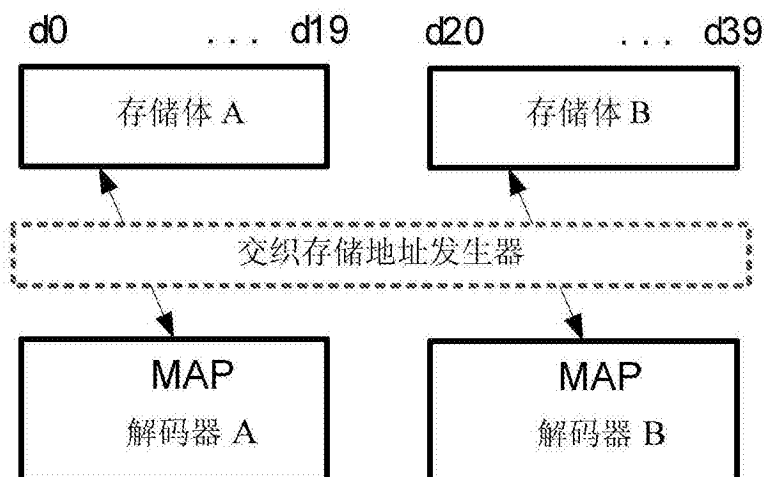


图3b

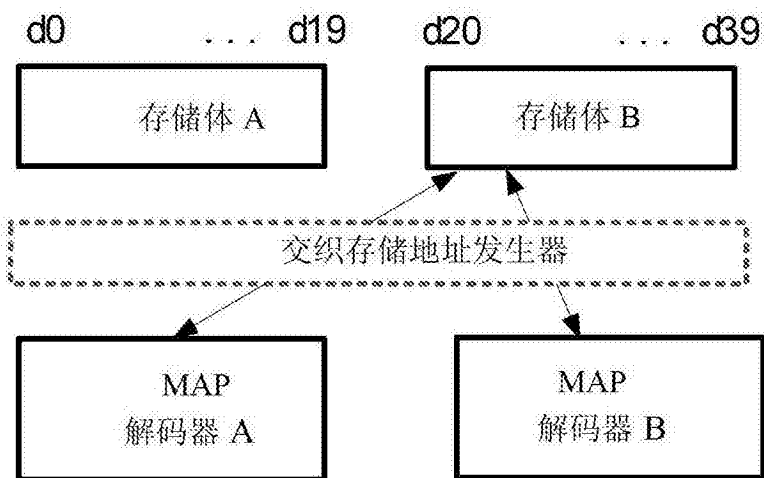


图3c

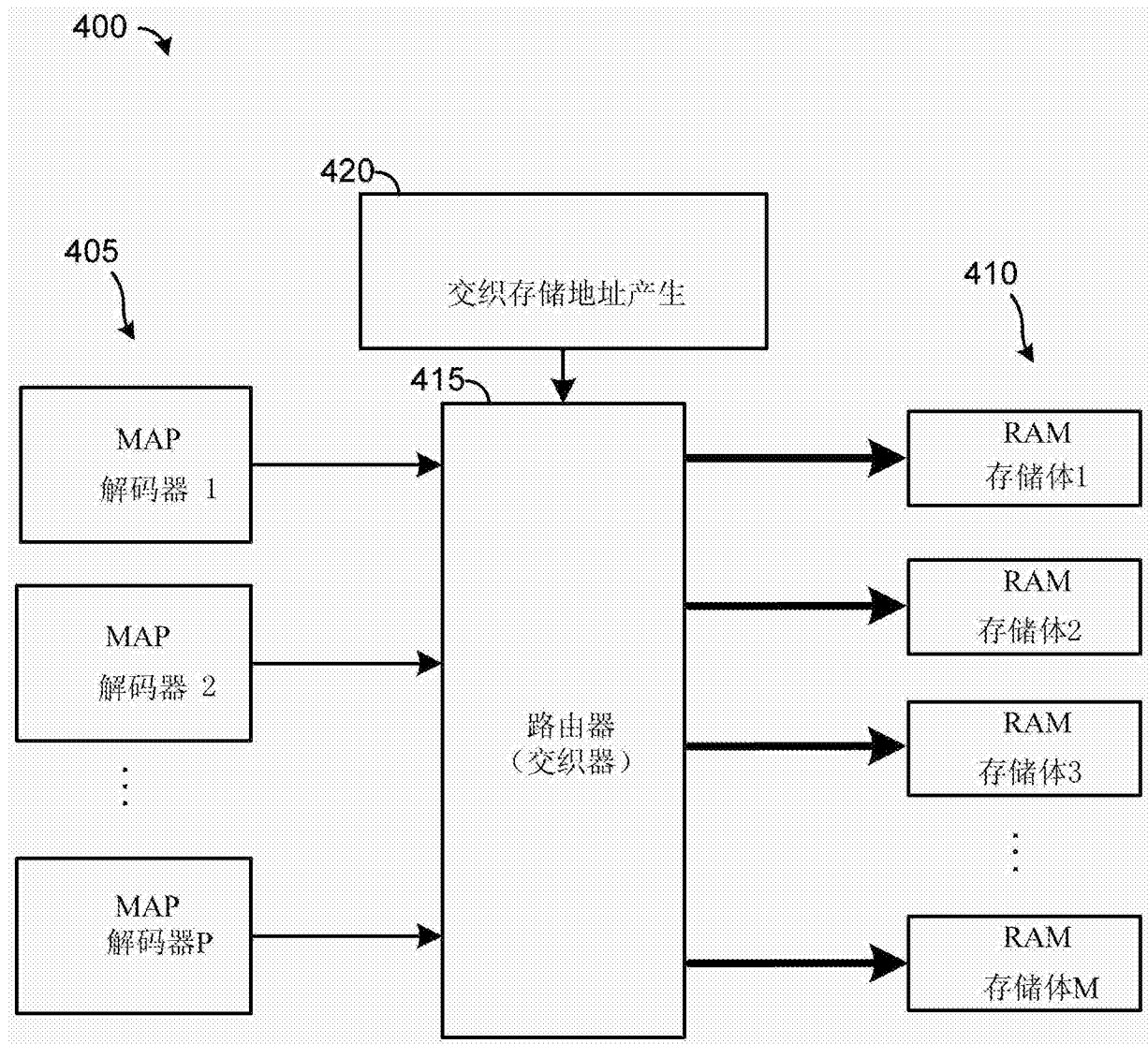


图4

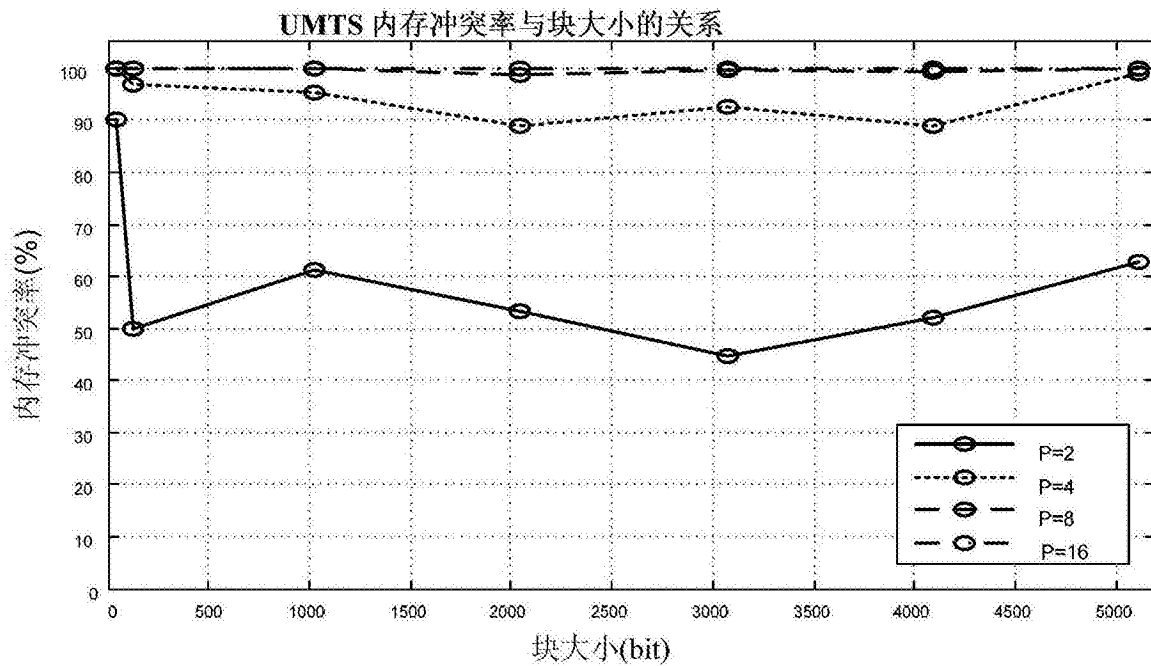


图5a

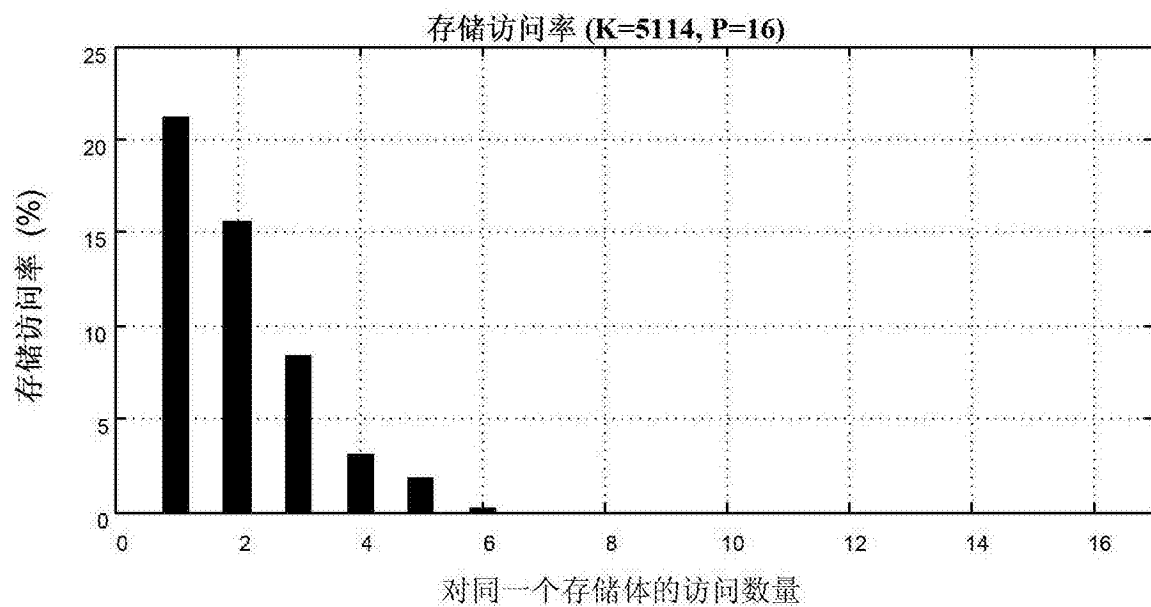


图5b

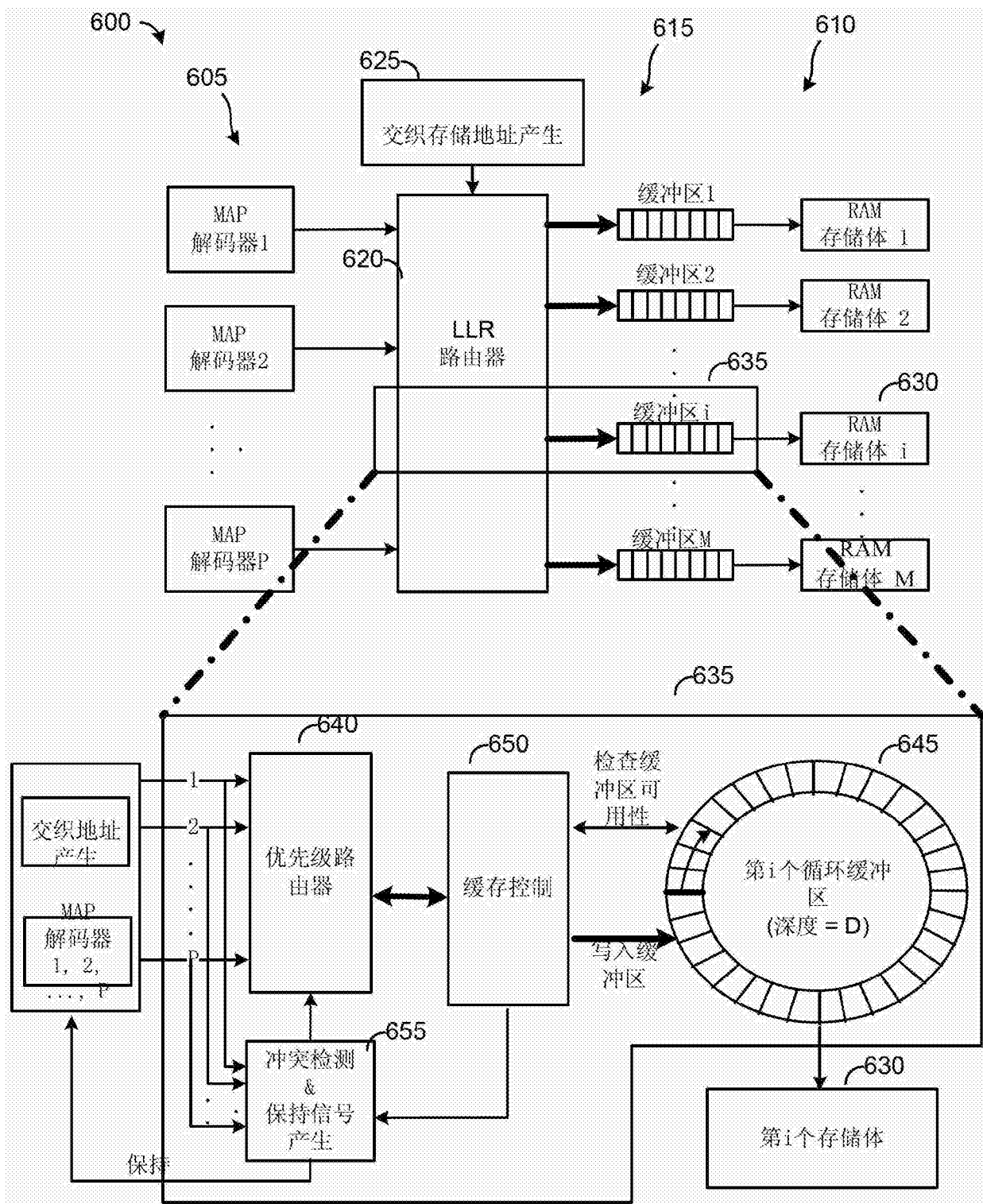


图6

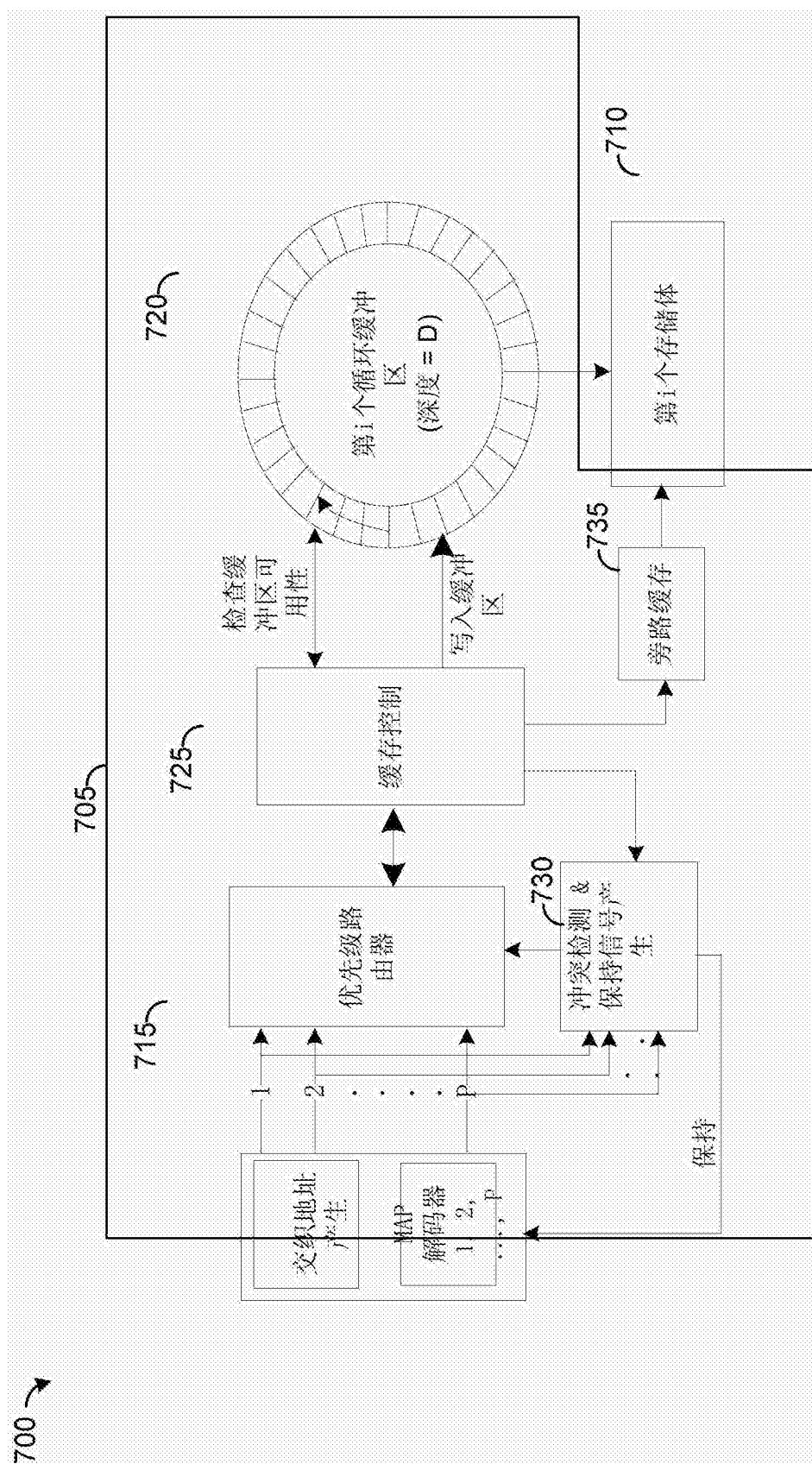


图7

		1	2	3	π 时钟周期)
子块 1	0	0	2	4	
	1279	1	3	5	
		1278	1276	1274	
子块 2	1280	1280	1282	1284	
	2559	1281	1283	1285	
		2558	2556	2554	
子块 3	2560	2560	2562	2564	
	3839	2561	2563	2565	
		3838	3836	3834	
子块 4	3840	3840	3842	3544	
	5119	3841	3843	3845	
		5118	5116	5114	

图8a

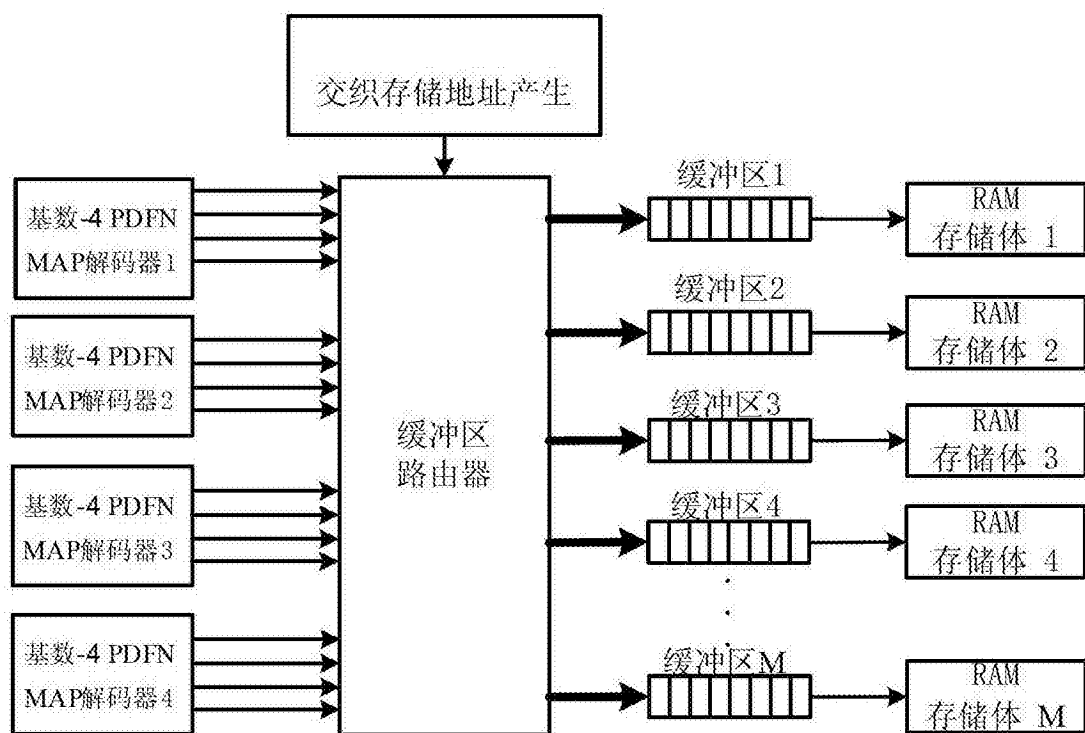


图8b

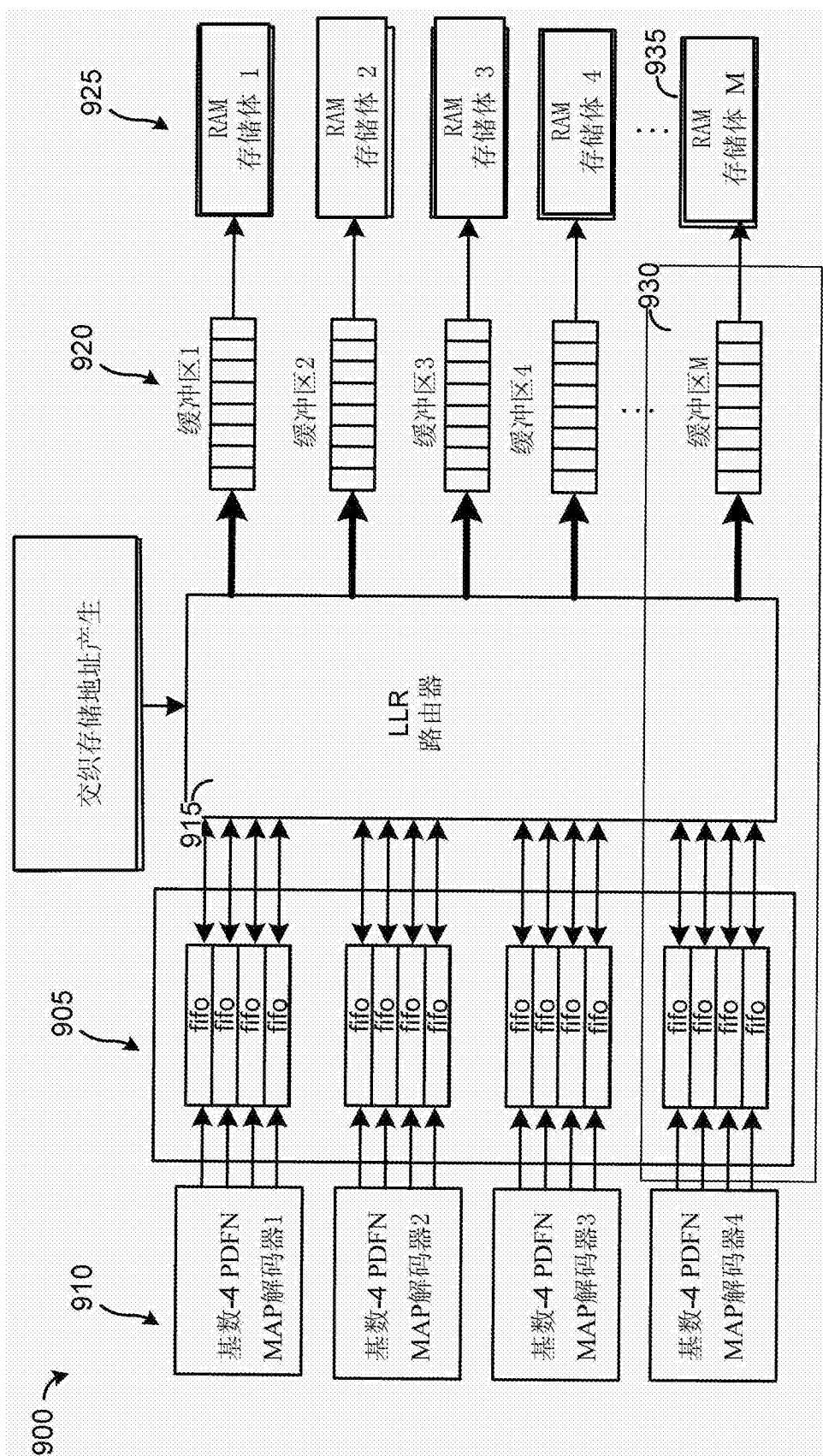


图9

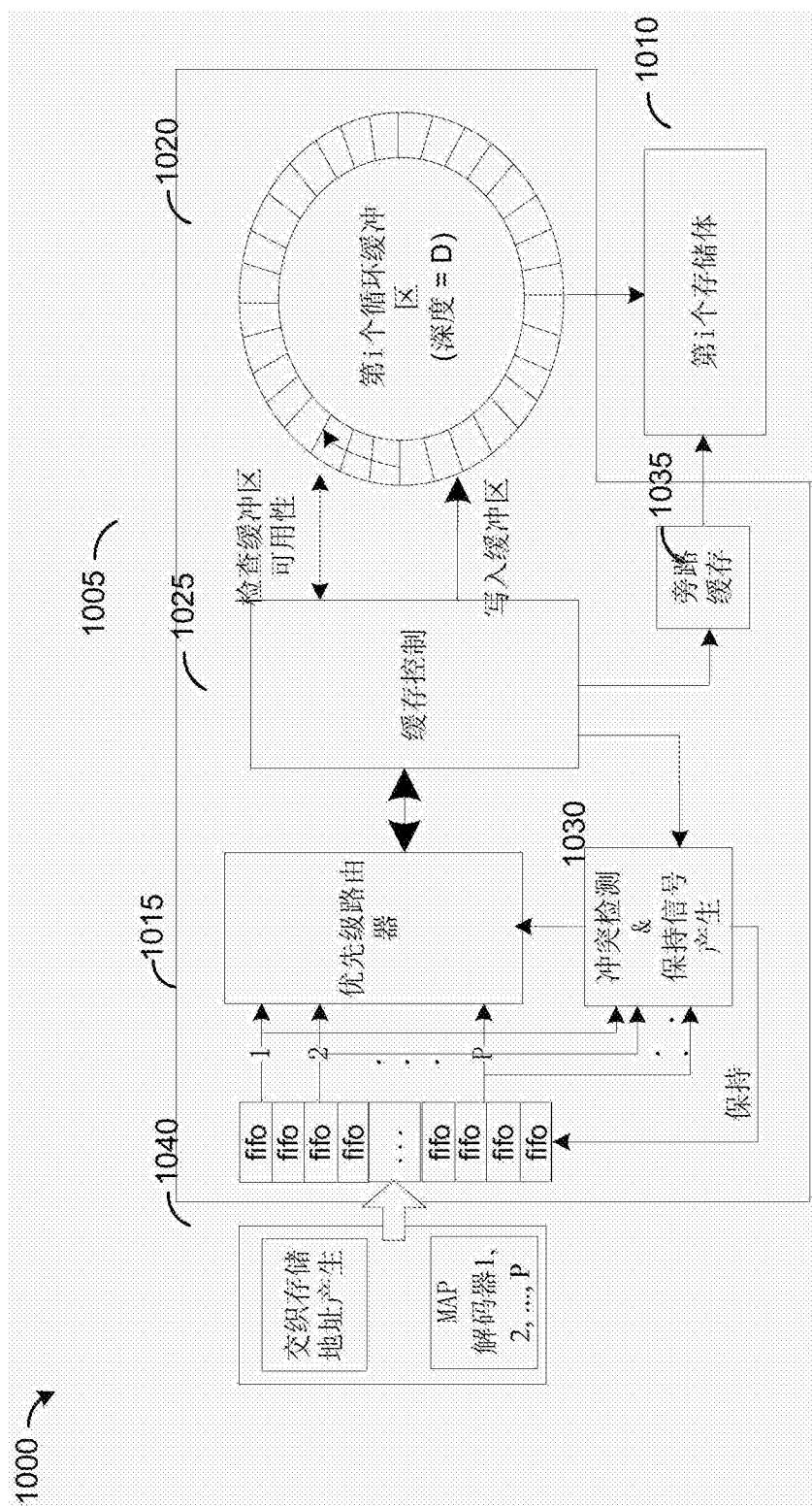


图10

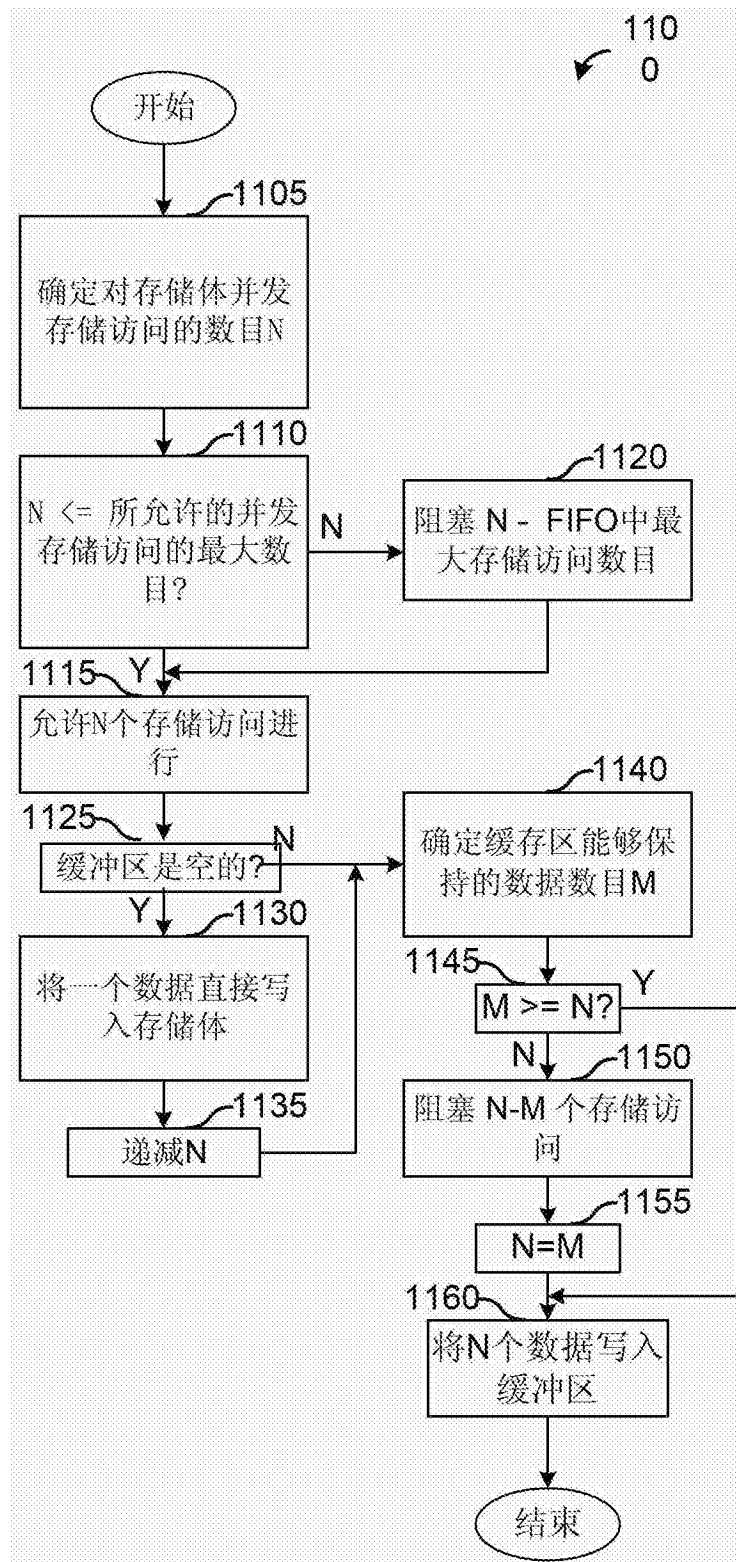


图11