



(12)发明专利

(10)授权公告号 CN 105765623 B

(45)授权公告日 2020.04.07

(21)申请号 201480045179.0

(22)申请日 2014.08.06

(65)同一申请的已公布的文献号

申请公布号 CN 105765623 A

(43)申请公布日 2016.07.13

(30)优先权数据

1314263.3 2013.08.08 GB

A/00812 2013.11.06 RO

14/082,645 2013.11.18 US

14/082,396 2013.11.18 US

(85)PCT国际申请进入国家阶段日

2016.02.14

(86)PCT国际申请的申请数据

PCT/IB2014/002541 2014.08.06

(87)PCT国际申请的公布数据

W02015/019197 EN 2015.02.12

(73)专利权人 大卫·莫洛尼

地址 爱尔兰都柏林

专利权人 理查德·里士满 大卫·多诺霍

布兰登·巴里 科马克·布里克

奥维迪乌·安德烈·韦萨

(72)发明人 大卫·莫洛尼 理查德·里士满

大卫·多诺霍 布兰登·巴里

科马克·布里克

奥维迪乌·安德烈·韦萨

(74)专利代理机构 北京品源专利代理有限公司

11332

代理人 杨生平 钟锦舜

(51)Int.Cl.

G06T 1/20(2006.01)

G06T 1/60(2006.01)

G09G 5/36(2006.01)

G09G 5/397(2006.01)

审查员 李婉怡

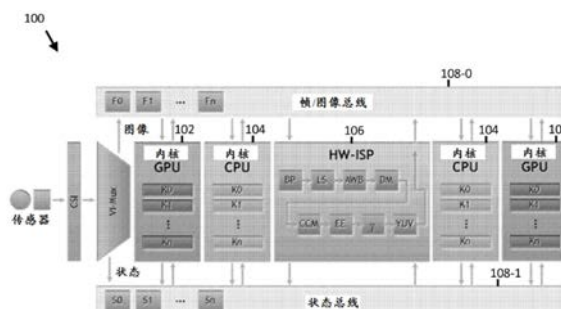
权利要求书4页 说明书26页 附图27页

(54)发明名称

计算成像管线

(57)摘要

本申请总体上涉及并行处理装置。所述并行处理装置可包括多个处理元件、存储器子系统和互连系统。所述存储器子系统可包括多个存储器片,所述多个存储器片中的至少一个与所述多个处理元件中的一个相关联,并且包括多个随机存取存储器(RAM)图块,每个图块具有单独的读取和写入端口。所述互连系统被配置来耦接所述多个处理元件和所述存储器子系统。所述互连系统包括局部互连和全局互连。



1. 一种并行处理装置,所述处理装置包括:

多个处理元件,其被构造来执行指令;

存储器子系统,其包括多个存储器片,所述存储器子系统包括与所述多个处理元件中的第一个相对应的存储器片中的第一个,其中所述存储器片中的第一个包括每个具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块;以及

多个仲裁块,所述多个仲裁块中的相应仲裁块与以下两者都相关联:(a) 所述多个处理元件中的第一个以及 (b) 所述多个RAM图块中的相应RAM图块,所述多个仲裁块中的第一个用于:

响应于确定所述处理元件中的第一个的访问请求是针对所述存储器片中的第一个,向所述多个处理元件中的第一个发送访问授权消息;以及

响应于确定所述处理元件中的第一个的访问请求不是针对被包括在所述存储器子系统内的所述存储器片中的第一个,将来自所述多个处理元件中的第一个的请求路由到互连系统。

2. 如权利要求1所述的并行处理装置,其中所述多个仲裁块中的第一个被构造来以循环法方式将所述访问授权消息发送到所述多个处理元件中的第一个。

3. 如权利要求1所述的并行处理装置,其中所述多个仲裁块包括冲突检测器,所述冲突检测器被构造来监视对所述多个RAM图块中的相应RAM图块的存储器访问请求,以及确定所述多个处理元件中的两个或更多个是否试图同时访问所述多个RAM图块中的相应RAM图块。

4. 如权利要求3所述的并行处理装置,其中所述冲突检测器耦接至多个地址解码器,其中所述多个地址解码器中的相应地址解码器耦接至所述多个处理元件中的相应处理元件,并且被构造来确定所述多个处理元件中的一个是否试图访问与所述多个仲裁块中的第一个相关联的所述多个RAM图块中的相应RAM图块。

5. 如权利要求1所述的并行处理装置,其中所述多个处理元件包括向量处理器或硬件加速器中的至少一个。

6. 如权利要求5所述的并行处理装置,其还包括每个被构造来提供对所述多个存储器片中的相应存储器片的访问的多个存储器片控制器。

7. 如权利要求6所述的并行处理装置,其中所述互连系统包括被构造来提供在所述向量处理器中的至少一个与所述存储器子系统之间的通信的第一总线。

8. 如权利要求7所述的并行处理装置,其中所述互连系统包括被构造来提供在所述硬件加速器中的至少一个与所述存储器子系统之间的通信的第二总线系统。

9. 如权利要求8所述的并行处理装置,其中所述第二总线系统包括片地址请求过滤器,所述片地址请求过滤器被构造来通过从所述硬件加速器的相应一个接收存储器访问请求,以及通过向所述硬件加速器的相应一个授予对所述存储器子系统的访问,调解在所述硬件加速器中的至少一个与所述存储器子系统之间的通信。

10. 如权利要求1所述的并行处理装置,其中所述多个处理装置中的一个包括用于增加所述存储器子系统的吞吐量的缓冲区,其中在所述缓冲区中的元件数目大于用于检索来自所述存储器子系统的数据的循环的数目。

11. 一种用于操作并行处理系统的方法,所述方法包括:

提供包括第一处理元件的多个处理元件,其中所述多个处理元件中的每一个被构造来

执行指令；

提供包括多个存储器片的存储器子系统，所述多个存储器片包括与所述第一处理元件相对应的第一存储器片，其中所述第一存储器片包括每个具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块；

通过与 (a) 所述多个 RAM 图块中的一个以及 (b) 所述第一处理元件两者都相关联的仲裁块，从所述第一处理元件接收请求，响应于所述请求，所述仲裁块：

响应于确定所述第一处理元件的访问请求是针对所述第一存储器片，向所述第一处理元件发送访问授权消息；以及

响应于确定所述第一处理元件的访问请求不是针对被包括在所述存储器子系统内的所述第一存储器片，将来自所述第一处理元件的请求路由到互连系统。

12. 如权利要求 11 所述的方法，其还包括通过所述仲裁块，向所述多个处理元件发送多个授权消息，以便以循环法方式授权对所述多个处理元件中的一个的访问。

13. 如权利要求 11 所述的方法，其还包括：

通过所述仲裁块中的冲突检测器来监视对所述多个 RAM 图块中的相应 RAM 图块的存储器访问请求；以及

确定所述多个处理元件中的两个或更多个是否正试图同时访问所述多个 RAM 图块中的相应 RAM 图块。

14. 如权利要求 11 所述的方法，其中所述多个处理元件包括向量处理器或硬件加速器中的至少一个。

15. 如权利要求 14 所述的方法，其还包括提供被构造来提供对所述多个存储器片中的相应存储器片的访问的多个存储器片控制器。

16. 如权利要求 15 所述的方法，其还包括经由所述互连系统的第一总线系统提供在所述向量处理器中的至少一个与所述存储器子系统之间的通信。

17. 如权利要求 16 所述的方法，其还包括经由所述互连系统的第二总线系统提供在所述硬件加速器中的至少一个与所述存储器子系统之间的通信。

18. 如权利要求 17 所述的方法，其中所述第二总线系统包括片地址请求过滤器，所述片地址请求过滤器被构造来通过从所述硬件加速器中的相应一个接收存储器访问请求，以及通过向所述硬件加速器中的相应一个授予对所述存储器子系统的访问，调解在所述硬件加速器中的至少一个与所述存储器子系统之间的通信。

19. 一种电子装置，其包括：

多个处理元件，所述多个处理元件中的相应处理元件用于执行指令；

存储器子系统，其包括多个存储器片，所述存储器子系统包括与所述多个处理元件中的第一个相对应的存储器片中的第一个，其中所述存储器片中的第一个包括每个具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块；以及

互连系统，其用于耦接所述多个处理元件和所述存储器子系统，其中所述互连系统包括：

局部互连，其用于耦接所述存储器片中的第一个和所述多个处理元件中的相应第一个；以及

全局互连，其被构造来耦接所述存储器片中的第一个和所述多个处理元件中的剩余的

处理元件；

处理器，其用于执行模块来：

检索与数据处理过程相关联的流程图，其中所述流程图包括多个节点和连接所述多个节点中的两个或更多个的多个边缘，其中相应节点识别操作，并且相应边缘识别在所连接的节点之间的关系；以及

将所述多个节点的第一节点分配至所述多个处理元件的第一处理元件，以及

将所述多个节点的第二节点分配至所述多个处理元件的第二处理元件，从而使与所述第一节点和所述第二节点相关联的操作并行。

20. 如权利要求19所述的电子装置，其中所述流程图以可扩展标记语言 (XML) 格式来检索。

21. 如权利要求19所述的电子装置，其中所述模块用于基于在所述电子装置中的存储器子系统的过往表现，将所述多个节点的所述第一节点分配至所述第一处理元件。

22. 如权利要求21所述的电子装置，其中所述存储器子系统包括用于在时间段内计数存储器冲突数目的计数器，并且所述存储器子系统的所述过往表现包括由所述计数器测量的所述存储器冲突数目。

23. 如权利要求19所述的电子装置，其中所述模块用于将所述多个节点的所述第一节点分配至所述第一处理元件，同时所述电子装置在执行所述流程图的至少一部分。

24. 如权利要求19所述的电子装置，其中所述模块用于：

检索多个流程图，并且

将与所述多个流程图相关联的操作分配至所述电子装置中的多个处理元件中的一个。

25. 如权利要求19所述的电子装置，其中所述模块用于通过所述多个处理元件交错存储器访问以便减少存储器冲突。

26. 如权利要求19所述的电子装置，其中所述电子装置包括移动装置。

27. 如权利要求19所述的电子装置，还包括应用编程接口 (API) 来指定所述流程图。

28. 如权利要求19所述的电子装置，其中所述模块用于通过以下步骤向所述多个处理元件提供输入图像数据：

将所述输入图像数据划分成多个条；并且

将所述输入图像数据的所述多个条中的一个提供至所述多个处理元件中的一个。

29. 如权利要求28所述的电子装置，其中所述输入图像数据的所述多个条的数目与所述多个处理元件的数目相同。

30. 一种方法，其包括：

通过利用与并行处理装置通信的处理器执行指令，检索与数据处理过程相关联的流程图，其中所述流程图包括多个节点和连接所述多个节点中的两个或更多个的多个边缘，其中每个节点识别操作，并且每个边缘识别在所连接的节点之间的关系；

将所述多个节点的第一节点分配至第一处理元件，

将所述多个节点的第二节点分配至第二处理元件，从而使与所述第一节点和所述第二节点相关联的操作并行，

配置存储器子系统，其具有多个存储器片，所述多个存储器片中的相应存储器片包括与所述第一处理元件相关联的第一存储器片，其中所述第一存储器片包括具有单独的读取

和写入端口的多个随机存取存储器 (RAM) 图块;

通过互连系统耦接所述第一处理元件、所述第二处理元件和所述存储器子系统;

耦接所述第一存储器片和所述第一处理元件;以及

通过全局互连耦接所述第一存储器片和所述第二处理元件。

31. 如权利要求30所述的方法,还包括以可扩展标记语言 (XML) 格式检索所述流程图。

32. 如权利要求30所述的方法,还包括基于所述并行处理装置中的第一存储器片的过往表现,将所述多个节点的所述第一节点分配至所述第一处理元件。

33. 如权利要求32所述的方法,还包括在所述存储器子系统中的计数器处,计数在时间段内的所述第一存储器片中的存储器冲突数目,并且所述第一存储器片的过往表现包括在所述第一存储器片中的所述存储器冲突数目。

34. 如权利要求30所述的方法,还包括将所述多个节点的所述第一节点分配至所述第一处理元件,同时所述并行处理装置在操作所述流程图的至少一部分。

35. 如权利要求30所述的方法,其还包括通过所述处理元件交错对所述第一存储器片的存储器访问来减少存储器冲突。

36. 如权利要求30所述的方法,还包括使用与所述并行处理装置相关联的应用编程接口 (API) 来指定所述流程图。

37. 如权利要求30所述的方法,还包括将输入图像数据划分成多个条;并且

将所述输入图像数据的所述多个条中的一个提供至所述多个处理元件中的一个。

38. 如权利要求37所述的方法,其中所述输入图像数据的所述多个条的数目与所述多个处理元件的数目相同。

计算成像管线

[0001] 相关申请的交叉引用

[0002] 本申请要求2013年11月18日提交的题目为“APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING COMPUTATIONAL IMAGING PIPELINE”的美国专利申请号14/082,396的更早优先权日的权益,所述专利申请要求2013年11月6日提交的题目为“APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE”的罗马尼亚专利申请OSIM Registratura A/00812的优先权,并且要求2013年8月8日提交的题目为“CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE”的英国专利申请号GB1314263.3的优先权。本申请还要求2013年11月18日提交的题目为“APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE COMPUTATIONAL IMAGING PIPELINE”的美国专利申请号14/082,645的更早优先权日的权益,所述专利申请要求2013年11月6日提交的题目为“APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE”的罗马尼亚专利申请OSIM Registratura A/00812的优先权,并且要求2013年8月8日提交的题目为“CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE”的英国专利申请号GB1314263.3的优先权。所述申请案中的每一个的全部内容在此均以引用的方式并入到本文中。

技术领域

[0003] 本申请大体涉及适用于图像和视频处理的处理装置。

背景技术

[0004] 因为图像分辨率和帧速率较高,每秒以数百兆像素的高聚合像素速率是普通的,所以计算图像和视频处理在存储器带宽方面是非常高要求的。此外,由于该领域处于其相对初期,算法处于恒定通量。因此,由于对算法的改变可意味着不能够适配,所以难以完全以硬件实施它们。同时,只依靠处理器中的实施方式的软件方法是不现实的。因此,大体希望提供可适应处理器和硬件加速器的灵活架构/基础结构。

[0005] 同时,对于此类视频和图像处理的需求较大程度来自便携式电子装置(例如,平板计算机和移动装置),其中功率消耗是一个关键考虑因素。因此,一般需要灵活基础结构来耦接可编程多核处理器和硬件加速器与允许它们以对于便携式电子装置必需的低功率水平递送持续的数据传输速率的高带宽存储器子系统。

发明内容

[0006] 根据公开的主题,本发明提供用于提供可配置和可组合计算成像管线的设备、系统和方法。

[0007] 公开的主题包括并行处理装置。处理装置包括各自被配置来执行指令的多个处理元件,以及包括多个存储器片(包括与所述多个处理元件中的一个相关联的第一存储器片)

的存储器子系统。第一存储器片包括各自具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块。并行处理装置可包括被配置来耦接多个处理元件和存储器子系统的互连系统。互连系统可包括被配置来耦接第一存储器片和多个处理元件中的一个的局部互连,和被配置来耦接第一存储器片和剩余的多个处理元件的全局互连。

[0008] 在一些实施方案中,多个RAM图块中的一个与仲裁块相关联,其中所述仲裁块被配置来从多个处理元件中的一个接收存储器访问请求,以及向多个处理元件中的一个授予对多个RAM图块中的一个的访问。

[0009] 在一些实施方案中,仲裁块被配置来以循环法方式授予对多个RAM图块中的一个的访问。

[0010] 在一些实施方案中,仲裁块包括冲突检测器,所述冲突检测器被配置来监视对多个RAM图块中的一个的存储器访问请求,以及确定多个处理元件中的两个或更多个是否试图同时访问多个RAM图块中的一个。

[0011] 在一些实施方案中,冲突检测器耦接至多个地址解码器,其中所述多个地址解码器中的每一个耦接至所述多个处理元件中的一个,并且被配置来确定所述多个处理元件中的一个是否试图访问与仲裁块相关联的多个RAM图块中的一个。

[0012] 在一些实施方案中,多个处理元件包括至少一个向量处理器和至少一个硬件加速器。

[0013] 在一些实施方案中,并行处理装置包括各自被配置来提供对多个存储器片中的一个的访问的多个存储器片控制器。

[0014] 在一些实施方案中,互连系统包括被配置来提供在至少一个向量处理器与存储器子系统之间的通信的第一总线。

[0015] 在一些实施方案中,互连系统包括被配置来提供在至少一个硬件加速器与存储器子系统之间的通信的第二总线系统。

[0016] 在一些实施方案中,第二总线系统包括片地址请求过滤器,其被配置来通过从至少一个硬件加速器接收存储器访问请求,以及通过向至少一个硬件加速器授予对存储器子系统的请求,调解在至少一个硬件加速器与存储器子系统之间的通信。

[0017] 在一些实施方案中,多个处理装置中的一个包括用于增加存储器子系统的吞吐量的缓冲区,其中在所述缓冲区中的元件数目大于用于检索来自存储器子系统的数据的循环的数目。

[0018] 公开的主题包括用于操作并行处理系统的方法。所述方法包括提供包括第一个处理元件和第二处理元件的多个处理元件,其中所述多个处理元件中的每一个被配置来执行指令。所述方法还包括提供包括多个存储器片(包括与第一处理元件相关联的第一存储器片)的存储器子系统,其中所述第一存储器片包括各自具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块。所述方法还包括通过与多个RAM图块中的一个相关联的仲裁块经由互连系统的局部互连,从第一处理元件接收第一存储器访问请求。所述方法另外包括通过仲裁块经由全局互连,向第一处理元件发送第一授权消息,以便授权所述第一处理元件访问多个RAM图块中的一个。

[0019] 在一些实施方案中,所述方法还包括通过仲裁块经由互连系统的全局互连,从第二处理元件接收第二存储器访问请求;以及通过仲裁块经由所述全局互连,向第二处理元

件发送第二授权消息,以便授权第二处理元件访问多个RAM图块中的一个。

[0020] 在一些实施方案中,所述方法还包括通过仲裁块,向多个处理元件发送多个授权消息,以便以循环法方式授权对多个RAM图块中的一个的访问。

[0021] 在一些实施方案中,所述方法还包括通过在仲裁块中的冲突检测器,监视对多个RAM图块中的一个的存储器访问请求,以及确定多个处理元件中的两个或更多个是否试图同时访问多个RAM图块中的一个。

[0022] 在一些实施方案中,多个处理元件包括至少一个向量处理器和至少一个硬件加速器。

[0023] 在一些实施方案中,所述方法还包括提供各自被配置来提供对多个存储器片中的一个的访问的多个存储器片控制器。

[0024] 在一些实施方案中,所述方法还包括经由互连系统的第一总线系统提供在至少一个向量处理器与存储器子系统之间的通信。

[0025] 在一些实施方案中,所述方法还包括经由互连系统的第二总线系统提供在至少一个硬件加速器与存储器子系统之间的通信。

[0026] 在一些实施方案中,第二总线系统包括片地址请求过滤器,其被配置来通过从至少一个硬件加速器接收存储器访问请求,以及通过向至少一个硬件加速器授予对存储器子系统的访问,调解在至少一个硬件加速器与存储器子系统之间的通信。

[0027] 公开的主题包括电子装置。所述电子装置包括并行处理装置。处理装置包括各自被配置来执行指令的多个处理元件,以及包括多个存储器片(包括与所述多个处理元件中的一个相关联的第一存储器片)的存储器子系统。第一存储器片包括各自具有单独的读取和写入端口的多个随机存取存储器(RAM)图块。并行处理装置可包括被配置来耦接多个处理元件和存储器子系统的互连系统。互连系统可包括被配置来耦接第一存储器片和多个处理元件中的一个的局部互连,和被配置来耦接第一存储器片和剩余的多个处理元件的全局互连。电子装置还包括处理器,其与并行处理装置通信,被配置来运行存储在存储器中的模块。所述模块被配置来:接收与数据处理过程相关联的流程图,其中所述流程图包括多个节点和连接多个节点中的两个或更多个的多个边缘,其中每个节点识别操作,并且每个边缘识别在所连接的节点之间的关系;以及将多个节点中的第一节点分配至并行处理装置的第一处理元件,并且将多个节点中的第二节点分配至并行处理装置的第二处理元件,从而并行化与第一节点和第二节点相关联的操作。

[0028] 在一些实施方案中,所述流程图以可扩展标记语言(XML)格式来设置。

[0029] 在一些实施方案中,所述模块被配置来基于在并行处理装置中的存储器子系统的过往表现,将多个节点中的第一节点分配至第一处理元件。

[0030] 在一些实施方案中,并行处理装置的存储器子系统包括被配置来在预先确定的时间段内计数存储器冲突数目的计数器,并且存储器子系统的过往表现包括由计数器测量的存储器冲突数目。

[0031] 在一些实施方案中,所述模块被配置来将多个节点中的第一节点分配至第一处理元件,同时并行处理装置操作流程图的至少一部分。

[0032] 在一些实施方案中,所述模块被配置来接收多个流程图,并且将与多个流程图相关联的全部操作分配至并行处理装置中的单个处理元件。

[0033] 在一些实施方案中,所述模块被配置来交错处理元件的存储器访问以便减少存储器冲突。

[0034] 在一些实施方案中,所述电子装置包括移动装置。

[0035] 在一些实施方案中,使用与并行处理装置相关联的应用编程接口 (API) 来指定流程图。

[0036] 在一些实施方案中,所述模块被配置来通过将输入图像数据划分成条并且将输入图像数据的一个条提供至多个处理元件中的一个,将输入图像数据提供至多个处理元件。

[0037] 在一些实施方案中,输入图像数据的条的数目与多个处理元件的数目相同。

[0038] 公开的主题包括方法。所述方法包括在与并行处理装置通信的处理器处,接收与数据处理过程相关联的流程图,其中所述流程图包括多个节点和连接多个节点中的两个或更多个的多个边缘,其中每个节点识别操作,并且每个边缘识别在所连接的节点之间的关系。所述方法还包括将多个节点中的第一节点分配至并行处理装置的第一处理元件,以及将多个节点中的第二节点分配至并行处理装置的第二处理元件,从而并行化与第一节点和第二节点相关联的操作。所述并行处理装置还包括存储器子系统,所述存储器子系统包括多个存储器片(包括与第一处理元件相关联的第一存储器片),其中所述第一存储器片包括各自具有单独的读取和写入端口的多个随机存取存储器 (RAM) 图块;以及被配置来耦接第一处理元件、第二处理元件和存储器子系统的互连系统。互连系统包括被配置来耦接第一存储器片和第一处理元件的局部互连,和被配置来耦接第一存储器片和第二处理元件的全局互连。

[0039] 在一些实施方案中,将多个节点中的第一节点分配至并行处理装置的第一处理元件包括基于并行处理装置中的第一存储器片的过往表现,将多个节点中的第一节点分配至第一处理元件。

[0040] 在一些实施方案中,所述方法还包括在存储器子系统中的计数器处,计数在预先确定的时间段内的第一存储器片中的存储器冲突的数目,并且第一存储器片的过往表现包括在第一存储器片中的存储器冲突的数目。

[0041] 在一些实施方案中,将多个节点中的第一节点分配至第一处理元件被执行,同时并行处理装置操作流程图的至少一部分。

[0042] 在一些实施方案中,所述方法还包括交错处理元件对第一存储器片的存储器访问以便减少存储器冲突。

[0043] 在一些实施方案中,使用与并行处理装置相关联的应用编程接口 (API) 来指定流程图。

[0044] 在一些实施方案中,所述方法还包括通过将输入图像数据划分成多个条并且将输入图像数据的多个条中的一个提供至多个处理元件中的一个,将输入图像数据提供至多个处理元件。

[0045] 在一些实施方案中,输入图像数据的多个条的数目与多个处理元件的数目相同。

附图说明

[0046] 当结合以下附图考虑时,参考本公开的主题的以下详细描述,可更完全地理解本公开的主题的各种目标、特征和优点,在附图中相同的参考数字指示相同的元件。

- [0047] 图1是图1描述Chimera的计算成像平台。
- [0048] 图2描述单元处理器的多核架构。
- [0049] 图3描述有效的低功率微处理器 (ELM) 架构。
- [0050] 图4示出根据一些实施方案的改善的存储器子系统。
- [0051] 图5示出根据一些实施方案的并行处理装置的区段。
- [0052] 图6示出根据一些实施方案的图块控制逻辑中的集中冲突检测系统。
- [0053] 图7示出根据一些实施方案的图块控制逻辑中的分布冲突检测系统。
- [0054] 图8示出根据一些实施方案的用于向请求者报告冲突信号的仲裁块。
- [0055] 图9示出根据一些实施方案的循环定向的仲裁块。
- [0056] 图10示出根据一些实施方案的用于减少由于存储器访问仲裁的存储器访问延迟的机制。
- [0057] 图11示出根据一些实施方案的调度软件的应用。
- [0058] 图12提供根据一些实施方案的具有并行处理装置的系统的分级结构。
- [0059] 图13示出根据一些实施方案的有向无环图 (DAG) 或流程图的描述如何可用于控制并行处理装置的操作。
- [0060] 图14A-14B示出根据一些实施方案的通过编译器和调度程序的任务的调度和发布。
- [0061] 图15示出根据一些实施方案的实时DAG编译器的操作。
- [0062] 图16根据一些实施方案将由OpenCL调度程序产生的时间表与由本发明的在线DAG调度程序产生的时间表比较。
- [0063] 图17示出根据一些实施方案的用于同步处理器和/或过滤器加速器的操作的阻挡机制。
- [0064] 图18示出根据一些实施方案的具有不同类型的处理元件的并行处理装置。
- [0065] 图19示出根据一些实施方案的本发明的多核存储器子系统。
- [0066] 图20示出根据一些实施方案的连接矩阵 (CMX) 基础结构的单个片。
- [0067] 图21示出根据一些实施方案的加速器存储器控制器 (AMC) 交叉式架构。
- [0068] 图22示出根据一些实施方案的AMC交叉式端口控制器。
- [0069] 图23示出根据一些实施方案的使用AMC的读取操作。
- [0070] 图24示出根据一些实施方案的使用AMC的写入操作。
- [0071] 图25示出根据一些实施方案的并行处理装置。
- [0072] 图26示出根据一些实施方案的包括并行处理装置的电子装置。

具体实施方式

[0073] 在以下的描述中,关于本公开的主题的系统和方法以及在其中此类系统和方法可操作的环境等的多个具体细节被示出,以便提供本公开的主题的彻底理解。然而,对于本领域的技术人员将明显的是本公开的主题可在没有此类具体细节的情况下被实践,并且在本领域中是众所周知的某些特征不被详细描述,以便避免本公开主题的主题的复杂化。此外,应当理解以下提供的实例是示例性的,并且预期存在在本公开主题的范围内的其他系统和方法。

[0074] 互连此类不同的处理资源(例如,处理器和硬件加速器)的一个可能方式为使用如由Nvidia开发的Chimera的计算摄影引擎中概述的总线。图1示出Chimera的计算摄影引擎。Chimera计算摄影引擎100包括多个图形处理单元(GPU)核心102,所述多个图形处理单元(GPU)核心102经由平级(flat-level)总线基础结构108(例如,连接全部处理元件的单个层级总线系统)连接至多核ARM处理器子系统104和硬件(HW)图像信号处理(ISP)加速器106。Chimera计算摄影引擎通常被呈现为从程序员摘要基础GPU核心102、CPU 104和ISP块106的细节的软件框架。此外,Chimera计算摄影引擎100描述经由两个信息总线108-0、108-1通过它们的计算摄影引擎的数据流,第一总线108-0运送图像或帧数据,并且第二总线108-1运送与每个帧相关联的状态信息。

[0075] 使用如在Chimera中的平级总线基础结构可以是便宜的并且方便实施。然而,使用平级总线基础结构作为互连诸如GPU核心102、CPU 104和ISP块106的非均匀处理元件的装置(例如,各种类型的处理元件)可具有许多值得注意的缺点。第一,使用总线来互连计算资源意味着存储器可能在整个系统中对于每个中央处理单元(CPU)104、图形处理单元(GPU)102和/或图像信号处理器(ISP)块106局部分布。因此,存储器不能根据程序员希望实施的计算摄影管线要求在处理管线内灵活分配。这种灵活性缺乏可意味着某个图像或视频处理难以或不能实施,或实施方式在帧速率、图像质量或另外方面被限制。

[0076] 第二,使用平级总线基础结构也意味着不同的计算资源(CPU 104、GPU 102和ISP块106)必须争夺总线带宽。这种争夺需要仲裁,这减少可用总线带宽的数量。因此,逐渐减少的理论带宽可用于实际工作。带宽的减少可意味着处理管线在帧速率、图像质量和/或功率方面未能满足应用的性能要求。

[0077] 第三,在特定计算资源附近的不充足的存储器可意味着数据必须在与给定的GPU 102、CPU 104、或硬件ISP块106相关联的存储器与另一计算资源之间来回传输。这种局部性缺乏意味着可引发附加的总线带宽和仲裁间接费用。此外,局部性缺乏也意味着消耗附加的功率。因此,可能难以或不可能以特定的目标帧速率来支持特定的算法。

[0078] 第四,使用平级总线基础结构也可在从非均匀处理元件构造管线中混合困难,所述非均匀处理元件的每一个可具有不同的延迟特性。例如,GPU核心102被设计来通过运行多个重叠处理线程容许延迟以便处理对存储器(通常外部DRAM)的多个未解决的访问来覆盖延迟,然而正常的CPU 104和硬件ISP块106不被设计成容许延迟的。

[0079] 用于互连不同的处理资源的另一方式通过在图2中示出的由IBM开发的单元处理器架构来提供。单元处理器架构200包括对于每个处理器204可用的局部存储(LS)202,也已知为协同执行单元(SXU)。单元处理器200依赖共享时间的基础结构,并且指示存储器访问(DMA)206传输以便以编程方式调度在一个处理器的LS 202与另一个处理器的LS 202之间的数据传输。关于单元架构200的困难为程序员面对的用于提前几百个循环(由于在单元架构200中的高延迟)明确调度后台数据传输,以确保共享数据当被需要时对于每个处理器204可用的复杂性。如果程序员不明确调度后台数据传输,那么处理器204将停止,这将使性能降级。

[0080] 用于互连不同的处理资源的另一方式为使用共享的多核存储器子系统以便在多核处理系统中的处理器之间有效共享数据。这个共享的多核存储器子系统在有效的低功率微处理器(ELM)系统中使用。图3示出ELM系统。ELM系统300包括系统302,其为用于ELM系统

中的计算资源的主要物理设计单元。系综302包括松散耦接的一簇四个处理器304。所述一簇四个处理器304共享诸如系综存储器306的局部资源和到互连网络的接口。系综存储器306捕获靠近处理器304的指令和数据工作集,并且被库存以便允许局部处理器304和网络接口控制器同时访问它。在系综302内的每个处理器304被分配在系综存储器306中的优选库。由处理器304对其优选库的访问被优先化高于(并且将阻塞访问)由其他处理器和网络接口的访问。对于单个处理器304是隐私的指令和数据可存储在它的优选库中以便提供确定性的访问次数。控制对读取和写入端口的访问的仲裁器被偏置以便建立在处理器304与存储器库306之间的亲和性。这允许软件当访问可由多个处理器共享的数据时,做出关于带宽可用性和延迟的更强假设。

[0081] 然而,由于物理上较大的随机存取存储器(RAM)块,ELM架构300可消耗许多功率。此外,ELM架构300在处理器304之间存在许多数据共享处可承受较低的吞吐量。此外,不对在处理器304与硬件加速器之间的数据共享进行规定,这在某些情况下可在功率和性能方面是有利的。

[0082] 本公开涉及用于使多个处理器和硬件加速器能够与其他处理器和硬件加速器同时访问共享数据的设备、系统和方法。本公开提供用于同时访问共享数据,而不被具有对访问局部存储的较高亲和性(例如,较高优先权)的局部处理器阻塞的设备、系统和方法。

[0083] 本公开的设备、系统和方法提供超过现有的多核存储器架构的大量利益。现有的多核存储器架构每个处理器使用单个整体式RAM块,这可限制可访问数据的带宽。本公开的架构可提供用于以与使用单个整体式RAM块的现有多核存储器架构相比基本上更高的带宽来访问存储器的机制。本公开架构通过每个处理器例示多个物理RAM块,代替每个处理器例示单个较大RAM块来获得这个更高的带宽。每个RAM块可包括专用访问仲裁块和周围的基础结构。因此,在存储器子系统系统中的每个RAM块可独立于其他RAM块被系统中的多个处理元件访问,例如向量处理器、精简指令集计算(RISC)处理器、硬件加速器、或DMA引擎等。

[0084] 因为基于单个大型RAM实例的存储器库是比基于多个更小的RAM实例的存储器库更区域有效的,所以与使用单个大型RAM实例相比,使用多个小型RAM实例更有利在一定程度上违反直觉。然而,更小的RAM实例的功率耗散通常显著低于单个大型RAM实例。此外,如果单个大型物理RAM实例将获取与多实例RAM块相同的带宽,那么单个大型物理RAM实例将引发比由多个物理RAM实例构成的RAM基本上更高的功率。因此,至少从功率耗散观点,存储器子系统可得益于使用多个物理RAM实例而不是使用单个大型RAM实例。

[0085] 具有多个物理RAM实例的存储器子系统可在每个RAM访问的成本具有添加优势,例如更小RAM块的存储器访问时间或功率消耗通常比更大RAM块的存储器访问时间或功率消耗低很多。这是由于用于从RAM读取/写入数据的缩短的位线。此外,用于更小RAM块的读取和写入的访问时间也是较低的(由于与更短的位线相关联的减少的阻容(RC)时间常数)。因此,耦接至基于多RAM的存储器子系统的处理元件可能以更高的频率操作,这继而减少由于静态泄漏电流导致的静态功率。当处理器和存储器被隔离至功率域中时,这可以是特别有用的。例如,当给定处理器或过滤器加速器已经完成它的任务时,与所述给定处理器或过滤器加速器相关联的功率域可有利地被门控关断。因此,在本公开的架构中的存储器子系统在可用带宽和功率耗散方面具有优异的特性。

[0086] 此外,具有多个RAM实例(各自具有仲裁的访问)的存储器子系统可提供用于数据

在处理器与硬件加速器之间被共享的许多方式,而不用通过锁定RAM块来使RAM块专用于特定处理器。原理上,如果更大的RAM被细分成N个子块,那么可用数据带宽增加大约N倍。这基于以下假设:数据可被多个处理元件及时分割以便减少同时发生的共享(例如,访问冲突)。例如,当消费者处理器或消费者加速器从被生产者处理器或生产者加速器填充的数据缓冲区读取数据时,那么存在数据缓冲区的同时发生的共享,导致访问冲突。

[0087] 在一些实施方案中,本公开的架构可提供用于减少数据的同时发生的共享的机制。具体地,本公开的架构可负责经由静态存储器分配机制和/或动态存储器分配机制来减少同时发生的共享。例如,在静态存储器分配机制中,在程序开始之前,例如在程序编译阶段期间,数据被映射至存储器的不同部分,以便减少数据的同时发生的共享。另一方面,在动态存储器分配方案中,在程序执行期间数据被映射至存储器的不同部分。静态存储器分配机制提供用于将存储器分配至数据的可预测的机制,并且它在功率或性能方面不引发任何实质的间接费用。

[0088] 又如,本公开的架构可与在控制器(例如,监督RSIC处理器)或调解对被分割跨多个RAM块的数据结构的访问的一个或多个处理器上运行的运行时间调度程序结合使用。运行时间调度程序可被配置来交错在数据(例如,图像帧)的部分(例如,线或图块)上操作的不同处理元件的开始时间,以便减少对共享数据的同时访问。

[0089] 在一些实施方案中,运行时间调度程序可用硬件仲裁块来补充。例如,硬件仲裁块可被配置来经由被设计来减少停止的共享确定性互连通过处理器(诸如,向量处理器)调解共享存储器访问。在一些情况下,硬件仲裁块可被配置来执行循环定向的调度。循环定向调度可包括以时钟循环间隔调度资源使用,如与以任务级别间隔调度资源使用相反,其可能需要多个时钟循环。以时钟循环间隔调度资源分配可提供更高的性能。

[0090] 在其他实施方案中,运行时间调度程序可用多个硬件加速器来补充,所述多个硬件加速器的每一个可包括输入缓冲区和输出缓冲区以便存储数据。输入和输出缓冲区可被配置来吸收(或隐藏)在访问外部资源(诸如外部存储器)中的延迟方差。输入和输出缓冲区可包括先进先出(FIFO)缓冲区,并且FIFO缓冲区可包括足够数目的狭槽来存储足够量的数据和/或指令,以便吸收在访问外部资源中的延迟方差。

[0091] 在一些实施方案中,本公开的设备、系统和方法提供并行处理装置。并行处理装置可包括其中的每一个可执行指令的多个处理器,诸如并行处理器。并行处理装置也可包括多个存储器片,每个存储器片与并行处理装置中的一个h相关联,以及超过并行处理装置中的其他处理装置给予那个处理器优先访问。每个存储器片可包括多个RAM图块,其中每个RAM图块可包括读取端口和写入端口。在一些情况下,每个存储器片可被设置用于提供对相关存储器片的访问的存储器片控制器。处理器和RAM图块可通过总线彼此耦接。在一些情况下,总线可将任何处理器与任何存储器片耦接。适当地,每个RAM图块可包括用于授予对图块的访问的图块控制逻辑块。图块控制逻辑块有时被称为图块控制逻辑或仲裁块。

[0092] 在一些实施方案中,并行处理装置可能还包括被配置来执行预先得定的处理功能(例如,图像处理)的至少一个硬件加速器。在一些情况下,预先得定的处理功能可包括过滤操作。

[0093] 在一些实施方案中,至少一个硬件加速器可经由单独总线耦接至存储器片。单独总线可包括相关联的加速器存储器控制器(AMC),所述加速器存储器控制器被配置来从至

少一个硬件加速器接收请求,并且通过相关的存储器片控制器向硬件加速器授予对存储器片的访问。从而应当理解由硬件加速器采用的存储器访问路径可不同于由向量处理器采用的路径。在一些实施方案中,至少一个硬件加速器可包括内部缓冲区(例如,FIFO存储器)以便弥补访问存储器片中的延迟。

[0094] 在一些实施方案中,并行处理装置可包括主机处理器。主机处理器可被配置来经由主机总线与AMC通信。并行处理装置也可被设置具有应用编程接口(API)。API向向量处理器和/或硬件加速器提供高级别接口。在一些实施方案中,并行处理装置可与提供用于并行处理装置的指令的编译器结合操作。

[0095] 在一些情况下,所述编译器被配置来在主机处理器上运行,该主机处理器不同于处理元件,诸如向量处理器或硬件加速器。在一些情况下,编译器被配置来经由指明图像处理过程的图像/视频API 1206(图12)接收流程图。编译器可能还被配置来将流程图的一个或多个方面映射至处理元件中的一个或多个,诸如向量处理器或硬件加速器。在一些实施方案中,流程图可包括节点和边缘,其中每个节点识别一个操作,并且每个边缘识别在节点(例如,操作)之间的关系,诸如实施操作的顺序。编译器可被配置来将一个节点(例如,操作)分配至处理元件中的一个以便并行化流程图的计算。在一些实施方案中,所述流程图可能以可扩展标记语言(XML)格式来提供。在一些实施方案中,编译器可被配置来将多个流程图分配至单个处理元件。

[0096] 在一些实施方案中,并行处理装置可被配置来测量它的性能并且向编译器提供信息。因此,编译器可使用从并行处理装置接收的过往表现信息,以便确定对并行处理装置中的处理元件的当前任务分配。在一些实施方案中,表现信息可指示由处理装置中的一个或多个处理元件经历的访问冲突数目。

[0097] 在一些情况下,并行处理装置可在视频应用中使用,所述视频应用计算量非常大。为解决视频应用的计算需求,并行处理装置可配置其存储器子系统以便在存储器访问期间减少在处理单元之间的访问冲突。为此,如先前讨论的,并行处理装置可将整体式存储器库细分成多个物理RAM实例,代替使用整体式存储器库作为单个物理存储器块。通过这种细分,每个物理RAM实例可被仲裁用于读取和写入操作,从而通过存储器库中的物理RAM实例的数目增加可用带宽。

[0098] 在一些实施方案中,硬件循环定向仲裁也可提供多个流量类和可编程调度掩膜。多个流量类和可编程调度掩膜可使用运行时间调度程序来控制。硬件循环定向仲裁块可包括端口仲裁块,所述端口仲裁块可被配置来在循环法调度方案中分配单个共享资源的多个请求者。在循环法调度方案中,请求者(例如,处理元件)以从请求者接收请求的顺序被授予对资源(例如,存储器)的访问。在一些情况下,端口仲裁块可增加循环法调度方案以便弥补多个流量类。单个共享资源可包括RAM图块、共享寄存器、或向量处理器、过滤器加速器和RISC处理器可访问以便共享数据的其他资源。另外,仲裁块可允许用优先权向量或超优先权向量来重写循环法资源分配。优先权向量或超优先权向量可由运行时间调度程序提供,以便优先化如特定的感兴趣应用需要的某些流量类(例如,视频流量类)。

[0099] 在一些实施方案中,处理元件可包括处理器(诸如,向量处理器或流式传送混合架构向量引擎处理器)、硬件加速器和硬件过滤器算子中的一个或多个。

[0100] 图4示出具有存储器子系统的并行处理装置,所述并行处理装置根据一些实施方

案允许多个处理器(例如,流式传送混合架构矢量引擎(SHAVE)处理器)共享多端口存储器子系统。具体地,图4示出适合于处理图像和视频数据的并行处理装置400。处理装置400包括多个处理元件402(诸如处理器)。在图4的示例性配置中,处理装置400包括8个处理器(SHAVE 0 402-0-SHAVE 7 402-7)。每个处理器402可包括通过其可从存储器412加载数据以及将数据存储至存储器412的两个负载存储单元404、406(LSU0、LSU1)。每个处理器402也可包括可将指令加载至其中的指令单元408。在其中处理器包括SHAVE的特定实施方案,所述SHAVE可包括精简指令集计算机(RISC)、数字信号处理器(DSP)、超长指令字(VLIW)和/或图形处理单元(GPU)中的一个或多个。存储器412包括在本文中被称为连接矩阵(CMX)片的多个存储器片412-0……412-7。每个存储器片412-N与对应的处理器402-7相关联。

[0101] 并行处理装置400还可包括耦接处理器402和存储器片412的互连系统410。互连系统410在本文中被称为shave间互连(ISI)。ISI可包括通过其处理器402可读取数据或将数据写入至存储器片412中的任一个的任何部分的总线。

[0102] 图5示出根据一些实施方案的并行处理装置的区段。区段500包括单个处理器402-N、与单个处理器402-N相关联的存储器片412-N、耦接单个处理器402-N和其他存储器片(未示出)的ISI 410、以及用于仲裁在存储器片412-N中的图块与处理器402-N之间的通信的图块控制逻辑506。如在区段500中示出的,处理器402-N可被配置来直接访问与处理器402-N相关联的存储器片412-N,处理器402-N可经由ISI访问其他存储器片(未示出)。

[0103] 在一些实施方案中,每个存储器片412-N可包括多个RAM图块或物理RAM块502-0……502-N。例如,具有128kB大小的存储器片412-N可包括被组织成4kx32-位字的4个32kB单端口的RAM图块(例如,物理RAM元件)。在一些实施方案中,图块502也可被称为逻辑RAM块。在一些实施方案中,图块502可包括单个端口的互补金属氧化物半导体(CMOS)RAM。单个端口的CMOS RAM的优点为它通常在大部分半导体工艺中可用。在其他实施方案中,图块502可包括多端口的CMOS RAM。

[0104] 在一些实施方案中,每个图块502可与图块控制逻辑506相关联。图块控制逻辑506被配置来从处理器402接收请求,并且提供对相关图块502的单独读取和写入端口的访问。例如,当处理元件402-N希望访问RAM图块502-0中的数据时,在处理元件402-N将存储器数据请求直接发送至RAM图块502-0之前,处理元件402-N可将存储器访问请求发送至与RAM图块502-0相关联的图块控制逻辑506-0。存储器访问请求可包括由处理元件402-N请求的数据的存储器地址。随后,图块控制逻辑506-0可分析存储器访问请求,并且确定处理元件402-N是否可访问请求的存储器。如果处理元件402-N可访问请求的存储器,那么图块控制逻辑506-0可将访问授予消息发送至处理元件402-N,并且随后处理元件402-N可将存储器数据请求发送至RAM图块502-0。

[0105] 由于存在通过多个处理元件同时访问的可能性,在一些实施方案中,图块控制逻辑506可包括冲突检测器,所述冲突检测器被配置来检测在其中两个或更多个的处理元件(诸如,处理器或加速器)试图访问存储器片中的图块中的任何一个的实例。冲突检测器可针对试图的同时访问监视对每个图块502的访问。冲突检测器可被配置来向运行时间调度程序报告访问冲突已经发生并且需要解决。

[0106] 图6示出根据一些实施方案的图块控制逻辑中的集中冲突检测系统。冲突检测系统可包括集中仲裁块608,其包括多个冲突检测器604和多个一位热码地址编码器602。在一

些实施方案中,一位热码地址编码器602被配置来从处理元件402中的一个接收存储器访问请求,并且确定存储器访问请求是否针对与一位热码地址编码器602相关联的RAM图块502中存储的数据。每个冲突检测器604可耦接至一个或多个一位热码地址编码器602,所述一位热码地址编码器602还耦接至可访问与冲突检测器602相关联的图块502的处理元件402中的一个。在一些实施方案中,冲突检测器604可耦接至与特定RAM图块502相关联的全部一位热码地址编码器602。

[0107] 如果存储器访问请求是针对与一位热码地址编码器602相关联的RAM图块502中存储的数据,那么一位热码地址编码器602可向特定的RAM图块的冲突检测器604提供位值“1”;如果存储器访问请求不是针对与一位热码地址编码器602相关联的RAM图块502中存储的数据,那么一位热码地址编码器602可向特定的RAM图块的冲突检测器604提供位值“0”。

[0108] 在一些实施方案中,一位热码地址编码器602被配置来通过分析存储器访问请求的目标地址,确定存储器访问请求是否针对与一位热码地址编码器602相关联的RAM图块502中存储的数据。例如,当与一位热码地址编码器602相关联的RAM图块502被设计具有0x0000和0x00ff的存储器地址范围时,那么一位热码地址编码器602可确定存储器访问请求的目标地址是否落在0x0000和0x00ff的范围内。如果是这样的,存储器访问请求是针对与一位热码地址编码器602相关联的RAM图块502中存储的数据;如果不是,存储器访问请求不是针对与一位热码地址编码器602相关联的RAM图块502中存储的数据。在一些情况下,一位热码地址编码器602可使用范围比较块来确定存储器访问请求的目标地址是否落在与RAM图块502相关联的地址范围内。

[0109] 一旦冲突检测器604从全部一位热码地址编码器602接收位值,冲突检测器604可计数在接收的位值中的“1”的数目(例如,合计位值),以便确定是否存在当前请求对相同RAM图块502访问的多于一个的处理元件402。如果存在当前请求对相同RAM图块502访问的多于一个的处理元件,那么冲突检测器604可报告冲突。

[0110] 图7示出根据一些实施方案的图块控制逻辑中的分布冲突检测系统。分布冲突检测器系统可包括分布式仲裁器702,所述分布式仲裁器702包括多个冲突检测器704。分布冲突检测系统的操作基本上类似于集中冲突检测系统的操作。在这种情况下,冲突检测器704以分布式方式来安排。具体地,分布式仲裁器702可包括以串行方式安排的冲突检测器704,其中每个冲突检测器704仅耦接到与特定RAM图块502相关联的一位热码地址编码器602的子集。这种安排不同于集中冲突检测器系统,在所述集中冲突检测器系统中冲突检测器704耦接至与特定的RAM图块502相关联的全部一位热码地址编码器602。

[0111] 例如,当特定的RAM图块502可由64个处理元件402访问时,第一冲突检测器704-0可接收来自32个处理元件的存储器访问请求,并且第二冲突检测器704-1可接收来自剩余的32个处理元件的存储器访问请求。第一冲突检测器704-0可被配置来分析来自耦接至其自身的32个处理元件的一个或多个存储器访问请求,并且确定请求对特定RAM图块502-0的访问的耦接至其自身的32个处理元件的第一数目的元件。并行地,第二冲突检测器704-1可被配置来分析来自耦接至其自身的32个处理元件的一个或多个存储器访问请求,并且确定请求对特定RAM图块502-0的访问的耦接至其自身的32个处理元件的第二数目的元件。随后,第二冲突检测器704可添加第一数目和第二数目以便确定64个处理元件中的多少个请求对特定RAM图块502-0的访问。

[0112] 一旦冲突检测系统检测冲突,冲突检测系统可向请求者402发送停止信号。图8示出根据一些实施方案的用于向请求者报告冲突信号的仲裁块。更具体地,使用OR门来组合在冲突检测系统中的范围比较块的输出,以便向请求者产生停止信号。停止信号指示多于一个的处理元件正试图访问在与请求者相关联的存储器片内的相同物理RAM子块。在接收停止信号之后,请求者可停止存储器访问操作直到冲突已经清除。在一些实施方案中,冲突可由独立于程序代码的硬件来清除。

[0113] 在一些实施方案中,仲裁块可能以循环间隔来操作。在此类实施方案中,仲裁块以时钟循环间隔而不是以任务级别间隔来分配资源,所述任务级别间隔可包括多个时钟循环。此类循环定向的调度可改善系统的性能。仲裁块可能以硬件实施,以使得仲裁块可实时执行循环定向的调度。例如,在任何特定实例处,以硬件实施的仲裁块可被配置来分配用于下一个时钟循环的资源。

[0114] 图9示出根据一些实施方案的循环定向的仲裁块。循环定向的仲裁块可包括端口仲裁块900。端口仲裁块900可包括第一端口选择块930和第二端口选择块932。第一端口选择块930被配置来确定存储器访问请求中的哪一个(被识别作为客户端请求向量中的数位位置)被分配至片端口[0]用于访问耦接至片端口[0]的存储器片,然而第二选择块932被配置来确定客户端请求向量中的哪一个被分配至片端口[1]用于访问耦接至片端口[1]的存储器片。

[0115] 第一端口选择块930包括第一前导一检测器 (LOD) 902-0和第二LOD 902-1。第一LOD 902-0被配置来接收可包括多个数位的客户端请求向量。在客户端请求向量中的每个数位指示是否已经从与那个数位位置相关联的请求者接收消息访问请求。在一些情况下,客户端请求向量以“有效高”模式来操作。一旦第一LOD 902-0接收客户端请求向量,第一LOD 902-0被配置来检测在其处请求首次变成非零的数位位置(从左至右计数),从而识别对第一端口选择块930的第一存储器访问请求(从左至右计数)。并行地,客户端请求向量可被AND逻辑算子912掩蔽,以便使用由掩膜寄存器906和掩膜左移相器904产生的掩膜来产生掩蔽的客户端请求向量。掩膜寄存器906可由与掩膜寄存器906通信的处理器来设置,并且掩膜左移相器904可被配置来向左移相由掩膜寄存器906表示的掩膜。第二LOD 902-1可从AND逻辑算子912接收掩蔽的客户端请求向量,并且检测在掩蔽的客户端请求向量中的前导1。

[0116] 来自第一LOD 902-0和第二LOD 902-1的输出随后被提供至端口[0]胜者选择块908。端口[0]胜者选择块908还接收两个附加的输入:优先权向量和超优先权向量。端口[0]胜者选择块908被配置来基于输入的优先权,确定接收的存储器访问请求中的哪一个应被分配至片端口[0]。在一些实施方案中,输入的优先权可被如下排名:开始于具有最高优先权的超优先权向量,将掩蔽的LOD向量划分成优先权请求和非优先权请求的优先权向量,随后是具有最低优先权的未掩蔽的LOD向量。在其他实施方案中,其他的优先权可被指定。

[0117] 尽管第一端口选择块930可被配置来确定客户端请求向量是否可被分配至片端口[0],第二端口选择块932可被配置来确定客户端请求向量是否可被分配至片端口[1]。第二端口选择块932包括第一后续一检测器 (TOD) 912-0、第二TOD 912-1、掩膜寄存器914、掩膜右移相器916、端口[1]胜者选择块918和掩蔽的AND逻辑块920。TOD 912被配置来接收可包括多个数位的客户端请求向量,并且检测在其处向量首次变成非零的数位位置(从右向左

计数)。第二端口选择块932的操作基本上类似于第一端口选择块930,不同的是它从输入向量的右向左操作,从而使用后续一检测器912-0来选择在输入请求向量中的后续一。

[0118] 端口胜者选择块908、918的输出也被提供至相同的胜者检测块910,所述胜者检测块910被配置来确定相同的存储器访问请求是否已经赢得对片端口[0]和片端口[1]的访问。如果相同的客户端请求向量已经赢得对片端口[0]和片端口[1]的访问,那么相同的胜者检测块910选择一个或多个片端口来路由请求,并且将另一个端口分配至在输入向量中的下一个最高排名请求。这避免了将资源过度分配至特定请求,从而改善对竞争请求者的资源分配。

[0119] 端口仲裁块900的操作通过开始于32位客户端请求向量的左手侧工作,并且掩蔽的L0D 902-1输出第一掩蔽的请求向量的位置,如果这个掩蔽的请求向量不被更高的优先权输入经由优先权或超优先权向量取代,那么对应于L0D位置的请求者胜利并且被授予对端口[0]的访问。L0D位置也用于经由32位左移相器904来提前掩膜位置,并且也用于与端口1L0D分配比较,以便检查相同的请求者是否已经被给予对两个端口的访问,并且在这种情况下,端口中只有一个被授予,其中触发器被切换以便在依次相同的胜者检测的情况下授予在端口0和1之间的交替基础上的访问。在这种情况下,来自掩蔽的检测器902-1的L0D输出经由优先权向量中对应的一个位已被分配优先权,请求的客户端被授予对端口0的2个背对背循环访问。在掩蔽的客户端请求向量中不存在前导一以及无更高的优先权请求存在的情况下,未掩蔽的L0D胜利并且被分配对端口0的访问。在以上情况中任一个的情况下,超优先权向量中的1位将重写先前请求中的任一个,并且将对端口0的未限制访问授予请求者。

[0120] 在图的下部分中的逻辑开始于请求向量的右手侧,并且另外以与上部分相同的方式操作,所述上部分开始于请求向量的左手侧。在这种情况下,仲裁块的端口1的操作在优先权方面等与逻辑部分的端口0相同。

[0121] 在一些实施方案中,处理元件402可包括缓冲区以便减少由于存储器访问仲裁的存储器访问延迟。图10示出根据一些实施方案的用于减少由于存储器访问仲裁的存储器访问延迟的机制。在典型的存储器访问仲裁方案中,存储器访问仲裁块是管线,当将共享资源(诸如RAM图块502)分配至多个处理元件(例如,请求者)中的一个时,其导致固定间接费用仲裁惩罚。例如,当请求者402将存储器访问请求发送至仲裁块608/702时,请求者402花费至少4个循环来接收访问授予消息,因为它在以下步骤的每一个中花费至少一个循环:(1) 在一位热码地址编码器602处分析存储器访问请求,(2) 在仲裁块608/702处分析一位热码地址编码器602的输出,(3) 通过仲裁块608/702将访问授予消息发送至一位热码地址编码器602,以及(4) 通过一位热码地址编码器602将访问授予消息发送至请求者402。接着随后,请求者402必须将存储器数据请求发送至RAM图块502,以及从RAM图块502接收数据,其中的每一个花费至少一个循环。因此,存储器访问操作具有至少6个循环的延迟。这个固定的惩罚将减少存储器子系统的带宽。

[0122] 这个延迟问题可用在处理元件402中维持的存储器访问请求缓冲区1002来解决。例如,存储器访问请求缓冲区1002每个时钟循环可从处理元件接收存储器访问请求,并且存储接收的存储器访问请求直到它们准备被发送至存储器仲裁块608/702。生效的缓冲区1002使将存储器访问请求发送至存储器仲裁块608/702的速率和从存储器子系统接收数据的速率同步。在一些实施方案中,缓冲区可包括队列。在缓冲区1002中的元件的数目(例如,

缓冲区的深度)可大于用于检索来自存储器子系统的数据的循环的数目。例如,当RAM访问延迟为6个循环时,缓冲区1002中的元件数目可以是10。缓冲区1002可减少仲裁延迟惩罚,并且改善存储器子系统的吞吐量。通过存储器访问请求缓冲区,在原理上,高达100%的总存储器带宽可被分配在请求者之间。

[0123] 应当理解关于使用多个RAM实例的潜在问题是通过允许由多个处理元件对库内子实例的同时访问,可导致存储器争夺。

[0124] 本公开提供用于解决存储器争夺的至少两种方法。第一,注意软件设计,如稍后将描述的,以便通过在存储器子系统中仔细布置数据来避免存储器争夺和/或存储器冲突以便减少争夺和/或存储器冲突。此外,与并行处理装置相关联的软件开发工具可允许在软件设计阶段期间报告存储器争夺或存储器冲突。因此,通过响应于在软件设计阶段期间报告的存储器争夺或存储器冲突来改善数据布置,存储器争夺问题或存储器冲突问题可被校正。

[0125] 第二,如以下进一步描述的,在架构内的ISI块被配置来检测硬件中的端口冲突(争夺)并停止具有较低优先权的处理元件。例如,ISI块被配置来分析来自处理元件的存储器访问请求,服务存储器访问请求序列,以及根据优先权顺序路由存储器访问请求,以使得来自全部处理元件的全部数据读取和写入以优选权顺序来完成。

[0126] 针对处理元件的优先权顺序可能以多个方式来建立。在一些实施方案中,可能以系统设计时间来静态限定优先权顺序。例如,优先权顺序可被编码作为系统寄存器的复位状态,以使得当系统功率上电时,具有一组预先分配的优先权的系统上电。在其他实施方案中,优先权顺序可经由用户可编程寄存器来动态确定。

[0127] 在一些实施方案中,程序设计员可计划用于他们的软件应用的数据布置,以便减少对在存储器片内的存储器的共享子块的争夺。在一些情况下,数据布置的计划可由仲裁块来帮助。例如,仲裁块可检测存储器争夺,在优先权的基础上向与最高优先权任务相关联的处理元件授予对存储器的访问,停止正争夺的其他处理元件,以及通过处理来展开争夺过程直到所述争夺已经被解决。

[0128] 图11示出根据一些实施方案的调度软件的应用。在本申请中,调度软件可协调在处理管线内的3x3模糊内核的实施方式。调度软件可在运行时间中确定操作顺序,并且通过处理元件协调所述操作。用于管线的流程图1100包括元件1-元件5,1102-1110。元件1 1102可包括输入缓冲区1112、处理块1144和输出缓冲区1114。输入缓冲区1112和输出缓冲区1114可使用触发器来实施。在一些实施方案中,其他元件1104-1110中的每一个可具有基本上类似于元件1 1102的结构。

[0129] 在一些实施方案中,元件2 1104可包括可用3x3模糊过滤器来过滤输入的处理元件(例如,向量处理器或硬件加速器)。元件2 1104可被配置来从共享缓冲区1118接收输入,所述共享缓冲区1118临时维持元件1 1102的输出。为了向输入施加3x3模糊内核,元件2 1104可在它开始操作之前,从共享输入缓冲区1118接收至少3行数据。从而,可在RISC处理器1122上运行的SW调度程序1120可在向元件2 1104发送它可开始过滤操作的信号之前,检测正确数目的数据行包含在共享缓冲区1118中。

[0130] 在存在3行数据的初始信号之后,SW调度程序1120可被配置来在每个附加新行已经被添加至滚动的3线缓冲区1118时,向元件2 1104发送信号。除逐行同步之外,逐循环仲

裁和同步被执行用于管线中的每个元件。例如,元件1 1102可包括每个循环产生一个完整输出像素的硬件加速器。为了实现这个吞吐量,硬件加速器可保持输入缓冲区1112是充满的,所以处理块1114具有足够的数据来继续它的操作。这样,处理块1114可产生足够的输出来保持元件1102的吞吐量尽可能的高。

[0131] 在一些实施方案中,软件工具链可使用存储器子系统从分析软件程序来预测存储器冲突。软件工具链可包括基于图形用户界面(GUI)的集成开发环境(IDE)(例如,基于Eclipse的IDE),从所述IDE开发者可编辑代码,调用编译器、汇编器,以及当被需要时执行源级别调试。软件工具链可被配置来经由在多个处理器上运行的程序的动态分析使用系统模拟器来预测存储器冲突,所述系统模拟器建模全部处理、总线、存储器元件和外围设备。软件工具链也可被配置来在日志文件或显示装置中记录在不同的处理器或硬件资源上运行的不同程序是否试图同时访问存储器片的特定块。软件工具链可被配置来以逐循环基础记录。

[0132] 在一些实施方案中,管线1100也可包括一个或多个硬件计数器(例如,用于每个存储器实例的一个计数器),每次存储器冲突发生时,硬件计数器增加。这些计数器随后可被硬件调试器(例如,JTAG)读取,并且在屏幕上显示或被记录到文件中。由系统程序设计员随后分析日志文件可允许不同地调度存储器访问以便减少存储器端口冲突的可能性。

[0133] 对于IBM的单元架构(在图2中示出)的程序设计员的一个关键困难是提前数百个循环可编程地调度数据传输,以使得当向量处理器访问数据时,数据可被DMA控制并存储在本地存储(LS)中。本公开的架构的一些实施方案可通过处理在硬件中的访问的仲裁和调度以及在用户可读硬件计数器中的记录冲突来解决这个问题。这允许本公开的架构用于创建高性能视频/图像处理管线。

[0134] 图12提供根据一些实施方案的具有并行处理装置的系统的分级结构。系统1200可包括具有多个处理元件(诸如过滤器)的并行计算系统1202和在并行计算系统1204上运行的软件应用1204、用于使应用1204与并行计算系统1202交互作用的应用编程接口(API) 1206、编译用于在并行计算系统1202上运行的软件应用1204的编译器1208和用于控制在并行计算系统1202中的处理元件的操作的运行时间调度程序1210。

[0135] 在一些实施方案中,本公开的并行处理装置可被配置来结合管线描述工具(例如,软件应用) 1204操作,所述工具允许将图像处理管线描述成流程图。管线描述工具1204能够以独立于基础硬件/软件平台的灵活方式描述图像/视觉处理管线。具体地,由管线描述工具使用的流程图允许独立于可用于实施流程图的处理元件(例如,处理器和过滤器加速器资源)描述任务。所得的管线描述工具的输出可包括有向无环图(DAG)或流程图的描述。DAG或流程图的描述可能以合适的格式(诸如XML)存储。

[0136] 在一些实施方案中,DAG或流程图的描述可对于系统1200中的全部其他工具是可访问的,并且可用于根据DAG控制并行处理装置的控制操作。图13示出根据一些实施方案的DAG或流程图的描述如何可用于控制并行处理装置的操作。

[0137] 在计算装置的实际操作之前,用于并行处理装置1202的编译器1208可进行(1)流程图1306的描述以及(2)可用资源1302的描述,并且产生指示如何跨多个处理元件执行DAG的任务列表1304。例如,当任务不能在单个处理元件上执行时,编译器1208可跨多个处理元件拆分所述任务;当任务可在单个处理元件上执行时,编译器1208可将任务分配至单个处

理元件。

[0138] 在一些情况下,当任务将只使用处理元件能力的一部分时,编译器1208可能以顺序方式融合以及调度将在单个处理元件上执行的多个任务,直到可由处理元件支持的极限。图14A示出根据一些实施方案的通过编译器和调度程序的任务的调度和发布。使用编译器和调度程序来调度任务的优点为编译器和调度程序可基于由任务执行的操作来自动调度任务。这对于现有技术是很大优点,在所述现有技术中程序设计员必须手动确定用于在运行特定任务的一个处理元件或一组处理元件上运行的代码的时间表,包括何时通过DMA调度从周围设备到CMX、从CMX到CMX块、以及从CMX返回周围设备的数据传输。这是费力的和易错的任务,并且DFG的使用允许这个过程自动化,从而节约时间和增加产量。

[0139] 在计算装置的运行时间期间,运行时间调度程序1210可基于由编译器1208产生的任务列表1304,跨可用处理元件动态调度任务。运行时间调度程序1210可在多核系统中的主机RSIC处理器1306上操作,并且可跨处理元件(诸如多个向量处理器、过滤器加速器和直接存储器访问(DMA)引擎),使用来自硬件性能监视器和计时器1308的统计来调度任务。在一些实施方案中,硬件性能监视器和计时器1308可包括可由运行时间调度程序1210读取的停止计数器、CMX冲突计数器、总线循环计数器(ISI、APB和AXI)和循环计数器。

[0140] 在一些实施方案中,运行时间调度程序1210可基于来自硬件性能监视器和计时器1308的统计,将任务分配至可用处理元件。硬件性能监视器和计时器1308可用于增加处理元件的效率,或使用更少数目的处理元件执行任务,以便节约功率或允许并行计算其他任务。

[0141] 为此,硬件性能监视器和计时器1308可提供性能度量。性能度量可以是指示处理元件的活动级的数目。性能度量可用于控制用于执行任务的例示处理元件的数目。例如,当与特定处理元件相关联的性能度量大于预先确定的阈值时,那么运行时间调度程序1210可例示与特定处理元件相同类型的附加处理元件,从而在更多的处理元件上分配任务。又如,当与特定处理元件相关联的性能度量小于预先确定的阈值时,那么运行时间调度程序1210可移除与特定处理元件相同类型的例示的处理元件中的一个,从而减少执行某个任务的处理元件的数目。

[0142] 在一些实施方案中,运行时间调度程序1210可优先化处理元件的使用。例如,运行时间调度程序1210可被配置来确定任务是否应被优选分配至处理器或硬件过滤器加速器。

[0143] 在一些实施方案中,运行时间调度程序1210可被配置来改变在存储器子系统内的CMX缓冲区布置,以使得系统可遵守运行时间配置标准。运行时间配置标准可包括例如图像处理吞吐量(每秒帧数)、能量消耗、由系统使用的存储器数量、操作处理器数目和/或操作过滤器加速器的数目。

[0144] 输出缓冲区可能以几种方式中的一种布置在存储器中。在一些情况下,输出缓冲区在存储器中可以是物理连续的。在其他情况下,输出缓冲区可以是“成块的”或“成片的”。例如,输出缓冲区可被拆分成N个垂直条、其中N为分配至图像处理应用的处理器数目。每个条位于不同的CMX片中。由于每个处理器可局部访问输入和输出缓冲区,这个布置可有助于处理器。然而,因为此类布置可导致过滤器加速器的许多冲突,所以这种布置对于过滤器加速器可以是有害的。过滤器加速器通常从左向右处理数据。因此,全部过滤器加速器将通过访问图像的第一条来开始它们的处理,这将从开始导致许多冲突。在其他情况下,输出缓

冲区可以是交织的。例如,输出缓冲区可被拆分跨全部16个CMX片,具有预先确定的尺寸的交织。预先确定的尺寸可以是128位。因为跨CMX传播访问减少冲突的可能性,所以输出缓冲区的交织布置可有助于过滤器加速器。

[0145] 在一些实施方案中,缓冲区(诸如输入缓冲区或输出缓冲区)可基于其生产者和消费者是硬件和/或软件而被分配。消费者是更重要的,由于他们通常需要更多的带宽(过滤器通常读取多行以及输出一行)。硬件过滤器相应于缓冲区的布置被编程(它们支持连续的、交织的和成片的存储器寻址)。

[0146] 图14B示出根据一些实施方案的用于使用编译器和调度程序来自动调度任务的过程。编译器基于DAG确定待由并行处理装置执行的任务列表。在步骤1402中,运行时间调度程序被配置来接收所任务列表,以及将所述任务列表维持在单独队列中。例如,当所述任务列表包括(1)将由DMA执行的任务、(2)将由处理器执行的任务和(3)将由硬件过滤器执行的任务时,运行时间调度程序可将任务存储在3个单独队列中,例如第一队列用于DMA、第二队列用于处理器以及第三队列用于硬件过滤器。

[0147] 在步骤1404-1408中,当相关联的硬件部件变得对于新任务可用时,运行时间编译器被配置来将任务发布至相关联的硬件部件。例如,在步骤1404中,当DMA变得可用于执行新任务时,运行时间编译器被配置来使用于DMA的第一队列出列,并且向DMA提供出列的任务。类似地,在步骤1406中,当处理器变得可用于执行新任务时,运行时间编译器被配置来使用于处理器的第二队列出列,并且向处理器提供出列的任务。另外,在步骤1408中,当硬件过滤器变得可用于执行新任务时,运行时间编译器被配置来使用于硬件过滤器的第三队列出列,并且向硬件过滤器提供出列的任务。

[0148] 在一些实施方案中,运行时间调度程序1210可使用来自硬件性能监视器和计时器1308的计数器值来调整处理元件的使用,特别地在多于一个管线(例如,软件应用1204)在处理元件阵列上同时运行的情况,由于这些管线不需要被共同设计。例如,如果被分配至每个管线的有效总线带宽小于预期的,并且在访问CMX存储器中发生的冲突数目是大的,运行时间调度程序1210可使用这个信息通过修改从2个管线队列提取任务的顺序来交错2个管线的执行,从而减少存储器冲突。

[0149] 在一些实施方案中,DAG编译器可实时操作(例如,在线)。图15示出根据一些实施方案的实时DAG编译器的操作。实时DAG编译器1502可被配置来接收DAG的输入XML描述、可用处理元件的描述和任何用户限定的约束(诸如,处理器数目、帧速率、功率耗散目标等)。随后,实时DAG编译器1502可被配置来跨处理元件(包括例如,DMA引擎、处理器、硬件过滤器和存储器)调度DAG部件,以便确保DAG如指定的当被映射至系统资源时可满足用户限定的约束。在一些实施方案中,实时DAG编译器1502可确定在DAG中的任务是否可能以广度优先方式被并行执行。如果DAG的广度大于可用于并行执行任务的处理元件的数目(例如,可用处理功率的量小于DAG的平行度),实时DAG编译器1502可“折叠”任务,以使得在可用处理元件上顺序执行任务。

[0150] 图16根据一些实施方案将由OpenCL调度程序产生的时间表与由本发明的在线DAG调度程序产生的时间表比较。由本发明的调度程序1208/1502产生的时间表可消除在典型OpenCL时间表存在的冗余的副本和DMA传输。因为用于执行在DAG任务上的处理的GPU远离执行时间表的处理器,所以这些数据传输在OpenCL时间表内存在。在移动装置中使用的

典型应用处理器中,大数据块在执行时间表的处理器与进行处理的GPU之间来回传输。在本发明的设计中,全部处理元件共享相同的存储器空间,并且因此不需要来回复制,从而节约相当多的时间、带宽和功率耗散。

[0151] 在一些实施方案中,当任务将只使用处理元件能力的一部分时,运行时间调度程序1210也可被配置来以顺序方式融合以及调度将在单个处理元件上执行的多个任务,直到可由处理元件支持的极限,如在图14中示出的。

[0152] 在图像处理应用中,调度程序可被配置来通过将图像划分成条来在处理器中划分处理任务。例如,图像可被划分成预先确定宽度的垂直条或水平条。

[0153] 在一些实施方案中,调度程序可预先确定用于特定图像处理应用的处理器数目。这允许调度程序预先确定用于图像的条数目。在一些实施方案中,过滤操作可由串联的处理器来执行。例如,当存在由应用执行的5个软件过滤器时,处理器402可各自被配置来在第一时间实例同时执行第一软件过滤器,在第二时间实例同时执行第二软件过滤器等。这意味着计算负载在被分配至特定图像处理应用的处理器中被更均匀地平衡。这是因为处理器被配置来以相同的顺序同时执行所述相同列表的过滤器。

[0154] 当过多的处理器被分配至图像处理应用时,处理器可花费许多时间在硬件过滤器加速器上空载、等待以便完成任务。另一方面,当过少的处理器被分配至应用时,硬件过滤器加速器可花费许多时间空载。在一些实施方案中,运行调度程序1210可被配置来检测这些情况,以及相应地适配。在一些实施方案中,调度程序1210可被配置来将处理器过度分配至特定的图像处理应用,并且一旦处理器已经先于硬件过滤器加速器完成它的任务,允许处理器关断。

[0155] 在一些实施方案中,调度程序可使用阻挡机制来使处理元件同步,诸如硬件过滤器加速器和处理器。调度程序输出可包括命令流。这些命令可包括(1)用于处理元件(诸如硬件过滤器加速器和处理器)的开始命令和(2)阻挡命令。阻挡命令指示处理元件应从进行下一个命令集等待,直到组中的全部处理元件已经到达阻挡命令,即使处理元件中的一些已经实际上完成它们的任务。在一些实施方案中,调度程序可基于由处理元件执行的任务之间的相关性提供阻挡命令。

[0156] 图17示出根据一些实施方案的用于使处理元件的阻挡机制同步。命令流包括阻挡命令(1702、1712)和任务命令(1704、1706、1708、1710)。每个任务命令可与处理元件相关联,并且如在以下图中所指示,可能在不同时间完成任务命令。调度程序因此可插入阻挡命令1712,以使得处理元件不进行未来任务,直到阻挡命令1712被清除。这个阻挡机制可被认为是并行任务的时间管线。

[0157] 在一些实施方案中,使用中断信号1714以硬件实施阻挡机制。例如,调度程序可编程表明哪些处理元件属于一组的位掩膜。当处理元件完成分配的任务时,与相应的处理元件相关联的中断信号被断言。一旦与组中处理元件相关联的全部中断信号已经被断言,处理元件的控制器可接收指示全部处理元件已经达到阻挡命令的全局中断信号。

[0158] 在一些实施方案中,中断源可包括SHAVE向量处理器、RSIC处理器、硬件过滤器或外部事件。具体地,硬件过滤器可支持各种模式(包括其中输入/输出缓冲区包含帧的非圆形缓冲区模式),并且过滤器可被配置来在它已经处理整个输入帧或写入完整的对应输出帧时,发布单个中断。过滤器也是可编程的以便使用对于图像尺寸、缓冲区基础地址/线步

幅等合适的设置,在来自帧的线、片或图块上操作。

[0159] 关于复杂并行处理装置的一个重要挑战为如何编程在并行处理装置中的处理元件,特别对于在类似计算机资源和存储器的资源方面是非常功率敏感和缺乏的埋置系统。由于帧尺寸和速率是非常高的并且正逐年强烈增加,计算成像、视频和图像处理特别在埋置系统上的性能方面是要求非常高的。

[0160] 对于在本文中存在的这个问题的解决方案是提供应用编程接口 (API) 1206,所述应用编程接口 (API) 1206允许程序设计员在高级别写入应用,而不用密切了解多核处理器架构1202的细节。使用软件API 1206,程序设计员可迅速创建新的图像或视频处理管线,而不用了解实施方式的密切细节,因为程序员无需考虑功能以在可编程处理器上的软件还是以硬件来实施的细节。例如,模糊过滤器内核的实施方式被设置作为在一个或多个处理器或硬件加速器过滤器上运行的参考软件实施方式。程序设计员初始可使用软件模糊过滤器实施方式,并且可切换至使用软件过滤器而无需改变整体管线实施方式,因为ISI、AMC和CMX仲裁块处理哪个处理器和HW资源访问物理存储器块以及以哪个顺序,而不是程序设计员。

[0161] 尽管以上描述的多端口存储器方法对于在相同处理器之间以高带宽和低延迟共享存储器是充足的,但它对于与其他装置共享带宽不是理想的。这些其他装置可以是具有不同的延迟要求的硬件加速器和其他处理器,特别在非常高带宽应用(诸如计算视频和图像处理)中。

[0162] 本公开的架构可结合多端口存储器子系统使用,以便提供更高的带宽来支持来自具有高度确定性延迟要求的多个可编程VLIW处理器的更多同时访问,一大群可编程图像/视频处理硬件过滤器和总线接口,以便允许由常规主机处理器和外围设备的控制 and 数据访问。图18示出根据一些实施方案的具有不同类型的处理元件的并行处理装置。并行处理装置包括多个处理器1802和多个过滤器加速器1804,并且多个处理器1802和多个过滤器加速器1804可分别经由ISI 410和加速器存储器控制器 (AMC) 1806耦接到存储器子系统412。

[0163] AMC 1086的子系统和多核存储器 (CMX) 子系统412提供芯片上存储,从而有助于在用于特定图像/视频处理应用中的处理器1802以及硬件过滤器加速器1804上的软件中的低功率流数字信号处理。在一些实施方案中,CMX存储器412被组织到被组织成64位字的16片128kB(总共2MB)中。每个处理器1802可具有对存储器子系统412中的片的直接访问,以及对存储器系统412中的全部其他片的间接(更高的延迟)访问。处理器1802可使用CMX存储器412用于存储指令或数据,同时硬件过滤器加速器1804使用CMX存储器412以便存储数据。

[0164] 为了有助于在非均匀处理元件之间的数据共享,同时允许当用HW过滤器加速器1804访问共享CMX存储器412时,不容忍延迟的处理器1802实现高性能,HW过滤器加速器1804被设计成容忍延迟的。这通过给每个HW过滤器加速器(过滤器)1804提供使得计时更加弹性的局部FIFO以及用于共享访问CMX的交叉式开关来实现,从而使得ISI自由支持SHAVE间通信而没有来自HW过滤器加速器的争夺,如在图10中根据一些实施方案示出的。

[0165] 除外部离开端口冲突之外,还可能与进入的ISI 410端口访问冲突。如果多于一个的外部片试图在任何一个循环中访问相同的存储器片,那么端口冲突可发生。LSU端口到ISI互连端口的映射是固定的,所以在没有任何冲突的情况下,SHAVE 0 1802-0可能通过LSU端口0访问片2,并且SHAVE 11 (1702-11)可能通过LSU端口1访问片2。ISI矩阵可允许每

个循环传输8x2端口x64位的数据。例如,SHAVE N 1802可通过LSU端口0和1访问片N+1,并且全部8个SHAVE处理器可同时访问而不用任何停止。

[0166] 在一些实施方案中,存储器子系统412可被逻辑划分成片(块)。图19示出根据一些实施方案的本发明的多核存储器子系统。图19示出在AXI、AHB、SHAVE、ISI与CMX以及过滤器加速器、AMC与CMX之间的详细总线互连。图示出两个AMC输入和两个AMC示出端口,2个ISI输入和2个ISI输出端口,到L2高速缓存和互相排他(互斥)块的连接、以及用于寻址4个存储器图块和FIFO的内部读取/写入仲裁和源多路复用、以及到ISI和AMC的4个存储器块输出的输出目的地选择。

[0167] 每个片可连接到包括12个SHAVE、DMA、纹理管理单元(TMU)和到板载主机处理器的AHB总线接口的16个可能的ISI输入源中的两个。类似地,每个片具2个输出ISI端口,其允许片将数据发送至包括12个SHAVE、DMA、纹理管理单元(TMU)以及到板载主机处理器的AXI和AHB总线接口的16个可能目的地中的2个。在优选实施方式中,片包含具有继而连接至局部SHAVE处理器(2x LSU以及2x 64位指令端口)的输入仲裁块的4个物理RAM块、2个ISI输入端口、2个AMC输入端口和用于SHAVE间消息传送的FIFO、以及消息传送FIFO、L2高速缓存和互斥块。

[0168] 在从CMX片的输出路径上,到目的地选择块的输入连接至4个RAM实例,连同L2高速缓存和硬件互斥块。来自如在图20中示出的作为块2002的目的地选择块的输出连接至2个局部LSU端口和指令端口(SP_1和SP_0)以及2x ISI输出端口和2x AMC输出端口。2个ISI端口允许局部片连接至来自12个可能处理器、DMA、TMU AXI和AHB主机总线的两个目的地。处理器被设置有经由SHAVE间互连(ISI)的对存储器的访问,所述SHAVE间互连(ISI)连接至包含在多核存储器子系统片中的2x64位ISI输入和2x64位ISI输出端口。由ISI互连提供的高带宽、低延迟的确定性访问减少在处理器中的停止并且提供高计算吞吐量。

[0169] 图21示出根据一些实施方案的AMC交叉式架构。AMC交叉结构1806可被配置来将硬件图像处理过滤器1804连接至CMX多核存储器片412的AMC端口。AMC 1806可包括一个或多个片端口控制器2102,对于每个CMX片412优选一个控制器。片端口控制器2102继而连接至片地址请求过滤器(SARF) 2104。SARF 2104继而连接至AMC客户端(在这个实施方案中,AMC客户端为硬件图像处理加速器)。SARF 2104从过滤器加速器接收读取/写入请求,并且向它们提供请求或授予信号,从而从已被授予写入访问的SIPP块接收数据和地址,以及将读取数据提供至已被授予读取访问的那些。另外,SARF通过AXI主机总线向系统主机处理器提供AXI主控,所述系统主机处理器允许主机经由AMC交叉式开关询问(读取/写入)CMX存储器。

[0170] 在一些实施方案中,片端口控制器2102被设置在AMC 1806中,所述AMC 1806与相关联的CMX存储器片412中的2x读取和2x写入端口通信,如图21所示。从CMX存储器子系统412的过滤器加速器侧可见,每个硬件过滤器连接至在加速器存储器控制器(AMC)交叉结构1806上的端口。AMC 1806具有一对64位读取端口,以及一对64位写入端口,将它连接至CMX存储器412的每个片(在优选实施方式中存在总共2MB的16片)。经由在加速器内的读取或写入客户端接口以及局部缓冲区将图像处理硬件加速器连接至AMC 1806允许更宽松的延迟要求,从而留下可用于具有高度确定性定时的ISI和处理器的更多带宽,从而允许减少处理器的停止。

[0171] 图20示出根据一些实施方案的CMX基础结构的单个片。所述片包含:仲裁器和源多

路复用,其允许八个可能64位源中的多达4个访问在CMX片中的4个物理SRAM图块;共享的L2高速缓存;用于线程中相互排他的处理器间协商的共享的互斥硬件块;以及用于SHAVE间低带宽消息传送的64位FIFO。六个输入源为:连接来自图21中示出的ACM的对应的slice_port[1]和slice_port[0]端口的AMCout1和AMCout0;另外,2个ISI端口、ISInout1和ISInout0;2x LSU端口LSU_1和LSU_0;以及最终,当结合时允许从CMX读取128位指令的2x指令端口SP_1和SP_0。仲裁器和源多路复用产生读取/写入地址和64位数据,以便响应于来自8个输入源的优先权访问而控制4x SRAM图块。尽管64位SHAVE间通信FIFO的输入连接至仲裁器和源多路复用器,但它的输出仅由CMX片中的局部处理器可读。在实践中,处理器经由ISInout1和ISInout0端口以及将CMX、片和处理器连接在一起的CMX片外部的ISI基础结构,与彼此的消息传送FIFO通信。

[0172] 除在图20中示出的片的2个AMC端口中的每一个上的64个请求者之间的仲裁之外,附加的2:1仲裁器被设置以便在AMC端口1与0之间仲裁。这个2:1仲裁器的目的是防止2个AMC端口中的一个或另一个使全部AMC端口带宽饱和,这可导致在请求端口中的一个上的过度停止。这个增强在多个重端口带宽请求者的存在下提供更平衡的资源分配,以及从而用于整体架构的更高的持续吞吐量。类似地,2:1仲裁器出于类似的原因在2个处理器端口SP1和SP0之间仲裁。

[0173] 仲裁和多路复用逻辑也通过处理器、或直接地或经由到共享L2高速缓存的ISI、经由在严格的循环法基础上在16个可能源之间共享访问的第二仲裁级控制访问,其中一个64位端口连接在第二级仲裁器与16个CMX片中的每一个之间。类似地,相同的逻辑允许访问32个硬件互斥器,其用于在12个板载处理器与2x 32位RISC处理器(经由在ISI上的AHB和AXI总线连接)上运行的线程中的互相排他的处理器间协商。

[0174] 在优选实施方式中的优先权是SP_1和SP_0具有最高优先权,随后LSU_1和LSU_0,接下来ISInout1和ISInout0,以及最终AMCout1和AMCout0,并且最终FIFO具有最低的优先权。这个优先权分配的原因是SP_1和SP_0通过处理器控制对CMX的程序访问,并且如果下一个指令是不可用的,那么处理器将立即停止,随后是也将致使处理器停止的LSU_1和LSU_0;类似地ISInout1和ISInout0来自其他处理器并且如果数据不是立即可用的,那么将致使其他处理器停止。AMCout1和AMCout0端口将具有最低的优先权,因为它们具有内置FIFO并且从而可在停止之前容忍许多延迟。处理器FIFO只被需要用于低带宽处理器间消息传送,并且从而具有全部中最低的优先权。

[0175] 一旦仲裁器已允许访问到4个SRAM图块、L2高速缓存、互斥器和FIFO的多达4个源,来自六个读取数据源(包括4个SRAM图块、L2高速缓存和互斥器)的输出数据被选择和指向8个可能64位目的地端口中的多达4个;与片相关联的处理器上的4个(SP_1、SP_0、LSU_1和LSU_0);与ISI(ISInout1和ISInout0)相关联的2个;以及最终与AMC(AMCout1和AMCout0)相关联的2个。由于只有4个64位源需要被分配至8个目的地端口,在输出多路复用器中不需要优先次序。

[0176] 图22示出根据一些实施方案的AMC交叉式端口控制器。AMC交叉式端口控制器2202包括循环法仲裁器2204,所述循环法仲裁器2204将端口控制器2202连接至其请求已经经由处理器被过滤的加速器1804。仲裁器随后可将来自AMC客户端的有效请求推送到端口控制器FIFO上。在读取请求的情况下,对请求的响应(读取客户端ID和线索引)被推送到读取TX

ID FIFO上。返回的片端口数据和有效信号被输入到端口控制器读取逻辑中,所述端口控制器读取逻辑从Rd TX ID FIFO取出读取客户端ID和线索引,并且将对应的片端口读取数据和可用信号推送到从其它可由请求的AMC客户端读取的Rd数据FIFO上。在FIFO的CMX侧上,端口停止逻辑取来自FIFO的请求,并且将片端口控制提供至在相关联的CMX存储器片上的2x AMC输入端口。

[0177] 到CMX的读取和写入客户端接口的数目是单独可配置的。任何客户端可寻址CMX的任何片(或多个片)。在CMX中具有16个片的情况下,每片2个端口并且600MHz的系统时钟频率下,可供给至客户端的最大总数据存储器带宽为143GB/s:最大带宽=600MHz*(64/8)*2*16=1.536e11B/s=143GB/s。

[0178] 在800MHZ的更高时钟速率下,带宽上升至191GB/sec。AMC在连接到它的硬件加速器块的其客户端读取/写入接口上的同时访问之间仲裁。两个读取/写入访问的最大值可能以每片每个时钟循环被授予,从而在600MHz的系统时钟频率下给出8.9GB/s的最大片存储器带宽。客户端访问不限制于CMX地址空间。落在CMX地址空间之外的任何访问被转发至AMC的AXI总线主控。

[0179] 图23示出根据一些实施方案的使用AMC1806的读取操作。在这个图解中,从地址范围A0-3读取4个数据字。AMC客户端(例如,过滤器加速器1804)首先断言在端口控制器输入端上的请求。端口控制器2202通过发布授予信号(gnt)来响应,所述授予信号(gnt)继而致使客户端发出地址A0、A1、A2以及最终A3。对应的rindex值出现在对应于每个授予的时钟(c1k)上升沿上。可见在客户端侧上的定时当与索引地址和数据比较时可以是非常有弹性的,所述索引地址和数据是从CMX片到端口控制器的输出。在端口控制器的CMX侧上的确定性定时允许在对于延迟非常敏感的AMC客户端与处理器之间有效共享对CMX的访问,然而FIFO允许以及在AMC客户端中的局部存储允许定时在CMX存储器子系统412的AMC客户端(例如,过滤器加速器)侧上是高度可变的。

[0180] 图24示出根据一些实施方案的使用AMC 1806的写入操作。在定时图中,示出经由AMC将4个数据字传输至CMX。请求被AMC客户端发布,并且在以下的时钟(c1k)上升沿上,授予信号(gnt)升高从而经由AMC传输与地址A0相关联的数据字D0。gnt信号随后降低持续一次时钟循环,并且在下一个c1k的上升沿上升高持续2个时钟循环,从而允许D1和D2在gnt再次降低之前分别寻址A1和A2。在下一个上升c1k沿上,gnt再次升高从而允许数据字D3被传输至地址A3,因此req和gnt在下一个c1k上升沿上降低从而等待下一个读取/写入请求。

[0181] 结合图12的示例性安排使用的流图像处理管线(SIPP)软件框架提供用于使用CMX存储器412实施图像处理管线的灵活方法,所述CMX存储器412用于扫描线缓冲区、框架图块(框架的子区段)或事实上处于高分辨率的整体框架,将在封装线接合中的外部DRAM晶片附连至图像/视频处理晶片附接的基底。SIPP框架注意复杂性,诸如处理图像边界(像素复制)和圆形线缓冲区管理,从而使以软件(在处理器上)实施的ISP(图像信号处理)功能更简单和更通用。

[0182] 图25示出根据一些实施方案的并行处理装置400。并行处理装置400可包括存储器子系统(CMX)412、多个过滤器加速器1804和用于仲裁对存储器子系统412的访问的总线结构1806。存储器子系统(CMX)412被构造来允许多个处理元件402并行访问数据和程序代码而没有停止。这些处理元件402可包括例如SHAVE(流混合架构向量引擎)处理器、适当VLIW

(超长指令字)处理器、对数据和程序并行访问而没有停止或过滤器加速器。另外,存储器子系统(CMX)412可为主机处理器(未示出)制定规定,以便经由诸如AXI(未示出)的并行总线访问CMX存储器子系统412。在一些实施方案中,每个处理元件402每个循环可通过它的LSU端口读取/写入多达128位,并且每个循环通过它的指令端口读取多达128位程序代码。除分别用于处理器和过滤器加速器的ISI和AMC接口之外,CMX 412通过AHB和AXI总线接口提供对存储器的同时读取/写入访问。AHB和AXI为标准ARM并行接口总线,其允许使用共享总线基础结构1806来连接处理器、存储器和外围设备。CMX存储器子系统412可被配置来每个循环处理18x128位存储器访问的峰值。

[0183] 加速器1804包括可在SIPP软件框架1200中使用的硬件图像处理过滤器的集合。加速器1804可使得计算最密集的功能中的一些能够从处理元件1802卸载。图示出多个过滤器加速器1804可如何连接至执行地址过滤、仲裁和多路复用的AMC 1804。多个MIPI摄像机串行接口2502也可连接至AMC 1804,在优选实施方式中,总共12个MIPI串行通道以2个通道的6组连接。AMC 1804也连接至AXI和APB接口,以便在参考实施方式中允许2个系统RISC处理器经由AMC访问CMX存储器。图中的最终元件为CMX 412,AMC 1804仲裁对CMX 412的访问,从而允许通过多个硬件过滤器加速器1804同时访问在CMX存储器412中的物理RAM实例。参考过滤器加速器1804也被示出,在这种情况下5x52D过滤器内核,其包含fp16(类似IEEE754的16位浮点格式)算术管线、相关联的管线停止控制器、用于将线输入存储至fp16管线的线缓冲区读取客户端、用于存储来自fp16管线的输出的线开始控制输入和线缓冲区写入客户端。为允许加速器配合在SIPP框架内,它们需要对CMX存储器的高带宽访问,这由加速器存储器控制器(AMC)提供。

[0184] 在一些实施方案中,CMX存储器子系统412可被拆分成与其相邻的处理元件402相关联的128kB块或片用于高速、低功率访问。在片内,存储器被组织成许多更小的图块,例如3x32kB、1x16kB和2x8KB的独立SRAM块。物理RAM尺寸可被选择作为区域利用和配置灵活性的折衷。任何处理元件402可能以相同的延迟(3个循环)访问存储器子系统(CMX)412中任何处的物理RAM,但在局部处理器片之外的访问在带宽上被限制,并且将比访问局部片消耗更多的功率。一般来说,为减少功率消耗以及增加性能,处理元件402可在专用存储器片中局部存储数据。

[0185] 在一些实施方案中,每个物理RAM可以是64位宽的。如果多于一个的处理元件402试图访问单个物理RAM,冲突可发生从而导致处理器停止。CMX将在端口冲突之间自动仲裁并且确保无数据丢失。对于每个端口冲突,处理元件402停止持续一个循环从而导致更低的吞吐量。在CMX 412内的仔细数据布置(由程序设计员)的情况下,端口冲突可被避免并且处理器循环被更好地利用。

[0186] 在一些实施方案中,多个处理器被设置具有加速器和CMX存储器。

[0187] 从图25中示出的图像处理HW架构可见,每个过滤器加速器1804可包括至少一个AMC读取和/或写入客户端接口以便访问CMX存储器412。在AMC 1806上的读取和/或写入客户端接口的数目被适当配置。AMC 1806可包括到CMX存储器412的每个片中的一对64位端口。AMC 1806将请求从其客户端路由至适当的CMX片412(通过部分地址解码)。来自不同客户端的对相同片的同时请求可能以循环法方式来仲裁。从CMX 412返回的读取数据被路由回到请求的AMC读取客户端。

[0188] AMC客户端(加速器)1804将满32位地址呈现给AMC 1806。来自不映射至CMX存储器空间的客户端的访问被转发至AMC的AXI主控。来自不同的客户端的同时访问(在CMX存储器空间之外)以循环法方式来仲裁。

[0189] AMC 1806不限于向过滤器加速器1804提供CMX 412访问;任何硬件加速器或第三方元件如果其存储器接口被适当地适配到AMC上的读取/写入客户端接口,那么可使用AMC 1806来访问CMX和更宽广的平台存储器空间。

[0190] 硬件图像处理管线(SIPP)可包括过滤器加速器1804、仲裁块1806、MIPI控制2502、到CMX多端口存储器412的APB和AXI接口和连接、以及示例性硬件5x5过滤器内核。这种安排允许用于图像处理应用的多个处理器1802和硬件过滤器加速器1804共享由多个单端口RAM(随机存取存储器)物理块组成的存储器子系统412。

[0191] 使用单端口的存储器增加存储器子系统的功率和区域效率,但限制带宽。本发明的安排通过使用多个物理RAM实例和提供对其的仲裁访问以便服务多个源,允许这些RAM块表现成能够服务来自多个源(处理器和硬件块)的多个同时读取和写入请求的虚拟多端口存储器子系统。

[0192] 使用应用编程接口(API)和在应用级的数据划分是重要的,以便确保用于在处理器和过滤器加速器之间的物理RAM块或它们中的处理器的连接减少,并且从而递送至处理器和硬件的数据带宽增加用于给定的存储器子系统配置。

[0193] 在一些实施方案中,并行处理装置400可驻留在电子装置中。图26示出根据一些实施方案的包括并行处理装置的电子装置。电子装置2600可包括处理器2602、存储器2604、一个或多个接口2606和并行处理器装置400。

[0194] 电子装置2600可具有存储器2604,诸如计算机可读介质、闪存存储器、磁盘驱动、光驱、可编程只读存储器(PROM)和/或只读存储器(ROM)。电子装置2600可被配置具有一个或多个处理器2602,所述处理器2602处理指令以及运行可存储在存储器2604中的软件。处理器2602也可与存储器2604和接口2606通信,以便与其他装置通信。处理器2602可以是任何可适用处理器,诸如结合CPU、应用处理器和闪存存储器、或精简指令集计算(RISC)处理器的芯片上系统。

[0195] 在一些实施方案中,编译器1208和运行时间调度程序1210可能以存储在存储器2604中的软件实施,并且在处理器2602上操作。存储器2604可以是非暂时性计算机可读介质、闪存存储器、磁盘驱动、光驱、可编程的只读存储器(PROM)、只读存储器(ROM)、或任何其他存储器或存储器的组合。软件可在能够执行计算机指令或计算机代码的处理器上运行。处理器还可能以使用专用集成电路(ASIC)、可编程逻辑阵列(PLA)、现场可编程门阵列(FPGA)、或任何其他集成电路的硬件来实施。

[0196] 在一些实施方案中,编译器1208可在单独计算装置中通过接口2606结合电子装置2600来实施。例如,编译器1208可能结合电子装置2600在服务器中操作。

[0197] 接口2606可能以硬件或软件来实施。接口2606可用于从网络以及局部源接收数据和控制信息,诸如对电视机的远程控制。电子装置也可提供各种用户接口,诸如键盘、触摸屏、轨迹球、触摸板和/或鼠标。电子装置在一些实施方案中也可包括扬声器和显示装置。

[0198] 在一些实施方案中,在并行处理装置400中的处理元件可包括能够执行计算机指令或计算机代码的集成芯片。处理器还可能以使用专用集成电路(ASIC)、可编程逻辑阵列

(PLA)、现场可编程门阵列(FPGA)、或任何其他集成电路的硬件来实施。

[0199] 在一些实施方案中,并行处理装置400可被实施作为芯片上系统(SOC)。在其他实施方案中,并行处理装置中的一个或多个块可被实施作为单独的芯片,并且并行处理装置可封装在封装系统(SIP)中。在一些实施方案中,并行处理装置400可用于数据处理应用。数据处理应用可包括图像处理应用和/或视频处理应用。图像处理应用可包括图像处理过程(包括图像过滤操作),视频处理应用可包括视频解码操作、视频编码操作、用于检测视频中的运动或物体的视频分析操作。本发明的附加应用包括基于图像、物体或视频的序列和增强现实应用的机器学习和分类,包括其中游戏应用从多个摄像机视图(包括深度启用摄像机)提取几何结构,并且从根据其线框几何结构(例如经由点云)可被提取用于通过GPU的随后顶点着色的多个视图提取特征。

[0200] 电子装置2600可包括移动装置,诸如蜂窝电话。移动装置可使用多个访问技术和有线通信网络与多个无线电接入网络通信。移动装置可以是提供增强的能力(诸如文字处理、网络浏览、游戏、电子书能力、操作系统和全键盘)的智能电话。移动装置可运行操作系统,诸如Symbian OS、iPhone OS、RIM's Blackberry、Windows Mobile、Linux、Palm WebOS和Android。屏幕可以是可用于将数据输入至移动装置的触摸屏,并且屏幕可代替全键盘使用。移动装置可具有运行应用或与应用通信的能力,所述应用由通信网络中的服务器提供。移动装置可从网络上的这些应用接收更新或其他信息。

[0201] 电子装置2600也可涵盖许多其他装置,诸如电视(TV)、视频投影仪、机顶盒或机顶单元、数字视频录像器(DVR)、计算机、上网本、膝上型计算机、平板计算机和可与网络通信的任何其他音频/视觉设备。电子装置也可在其栈或存储器中保持全球定位坐标、配置文件信息、或其他位置信息。

[0202] 将理解尽管在本文中已经描述了几个不同的安排,但每个的特征可有利地以各种形成组合在一起以便获得优点。

[0203] 在先前描述的说明书中,已经参考具体示例描述本应用。然而,将显而易见的是:在不脱离如在所附权利要求书中阐述的本发明的更宽广精神和范围的情况下,可以对其做出各种修改和改变。例如,连接可以是适合于从相应的节点、单元或装置传输信号或将信号传输到相应的节点、单元或装置(例如,经由中中间装置)的任何类型的连接。相应地,除非暗示或另行指出,连接例如可以是直接连接或间接连接。

[0204] 应理解本文描绘的架构仅是示例性的,并且事实上实现相同功能性的许多其他架构可被实施。在抽象但仍确定的意义上,用于实现相同功能性的任何部件安排为有效“关联的”,以使得实现期望的功能性。从而,在本文中组合以便实现特定功能性的任何两个部件可被看做彼此“相关联”,以使得不考虑架构或中间部件,实现期望的功能性。同样地,这样相关联的任何两个部件也可被看做彼此“以操作方式连接”或“以操作方式耦接”,以便实现期望的功能性。

[0205] 此外,本领域的那些技术人员将认识到在以上描述的操作的功能性之间的边界仅是说明性的。多个操作的功能性可被组合成单个操作,和/或单个操作的功能性可被分布在附加的操作中。此外,替代性实施例可包括特定操作的多个实例,并且操作的顺序在各种其他实施方案中可改变。

[0206] 然而,其他修改、变化和替代方案也是可能的。因此,应在说明性意义而不是限制

性意义上理解本说明书和附图。

[0207] 在权利要求中,放置在括号之间的任何参考符号不得理解为是对权利要求的限制。词语“包括”不排除在权利要求中列出的那些之外的其他元件或步骤的存在。此外,如本文所使用的术语“一个(种) (a/an)”被定义为一个或多于一个。另外,在权利要求中使用诸如“至少一个”和“一个或多个”的引导短语不应被理解为暗示由不定冠词“a”或“an”引入的另一个权利要求元件将包含此类引入的权利要求元件的任何特定的权利要求限制于仅包含一个此类元件的发明,即使当相同的权利要求包括引导短语“一个或多个”或“至少一个”以及诸如“a”或“an”的不定冠词时。这对于定冠词的使用同样适用。除非另行指出,诸如“第一”和“第二”的术语用于在此类术语描述的元件之间进行任意辨别。从而,此类术语不一定旨在指示此类元件的时间或其他优先次序。在互相不同的权利要求中叙述某些测量的不争事实不指示这些测量的组合不可用于获得优点。

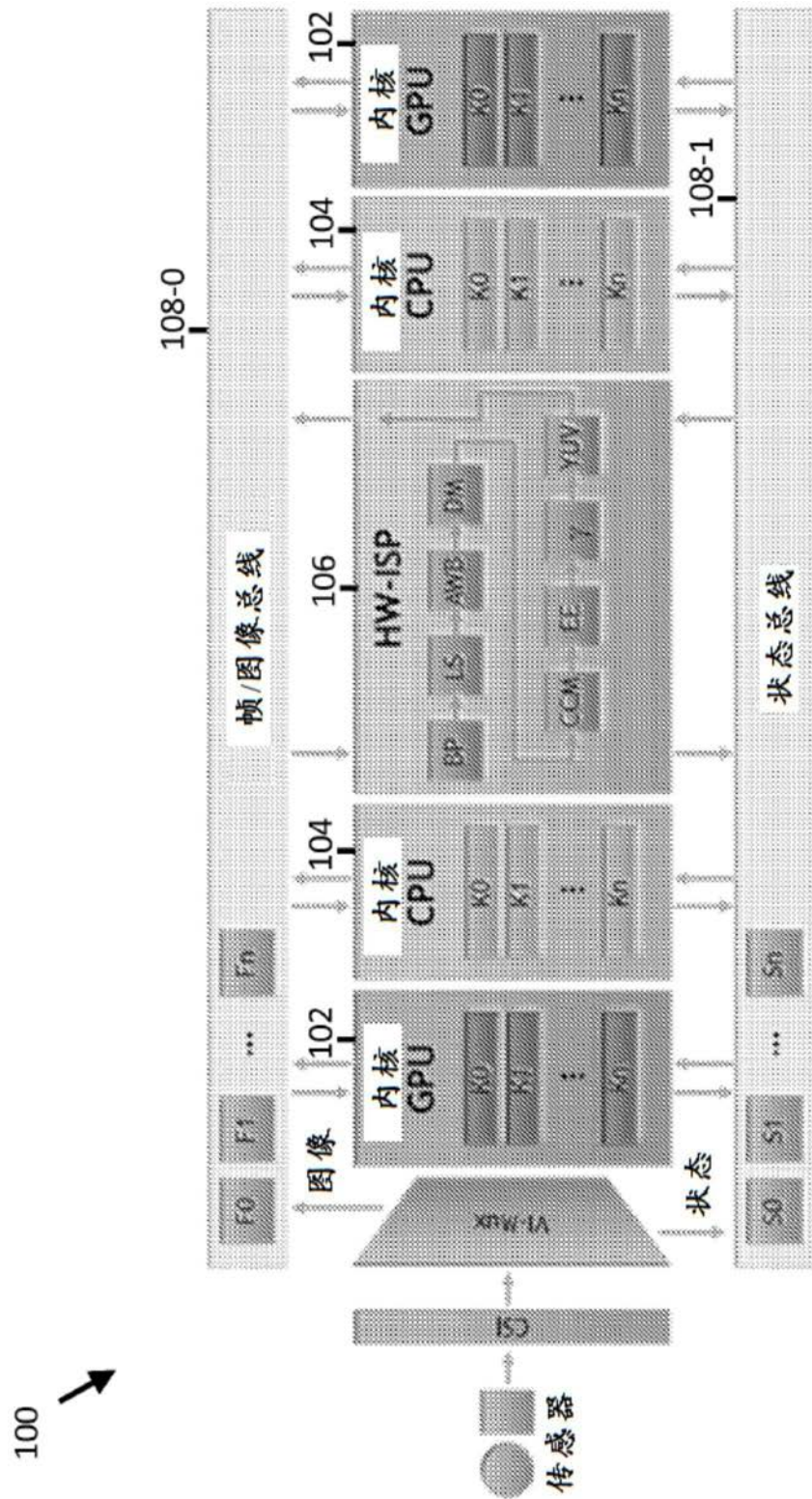


图1

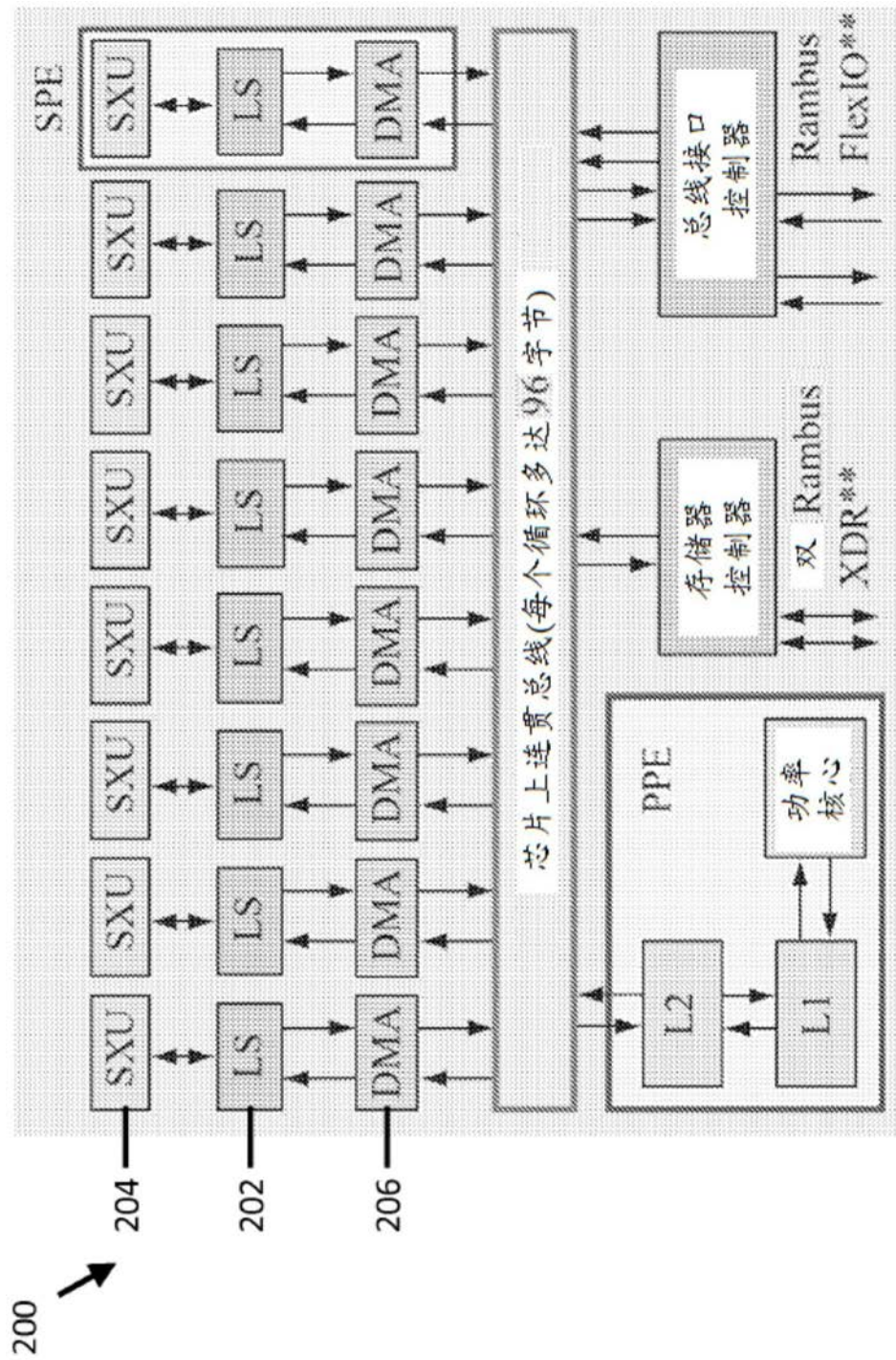


图2

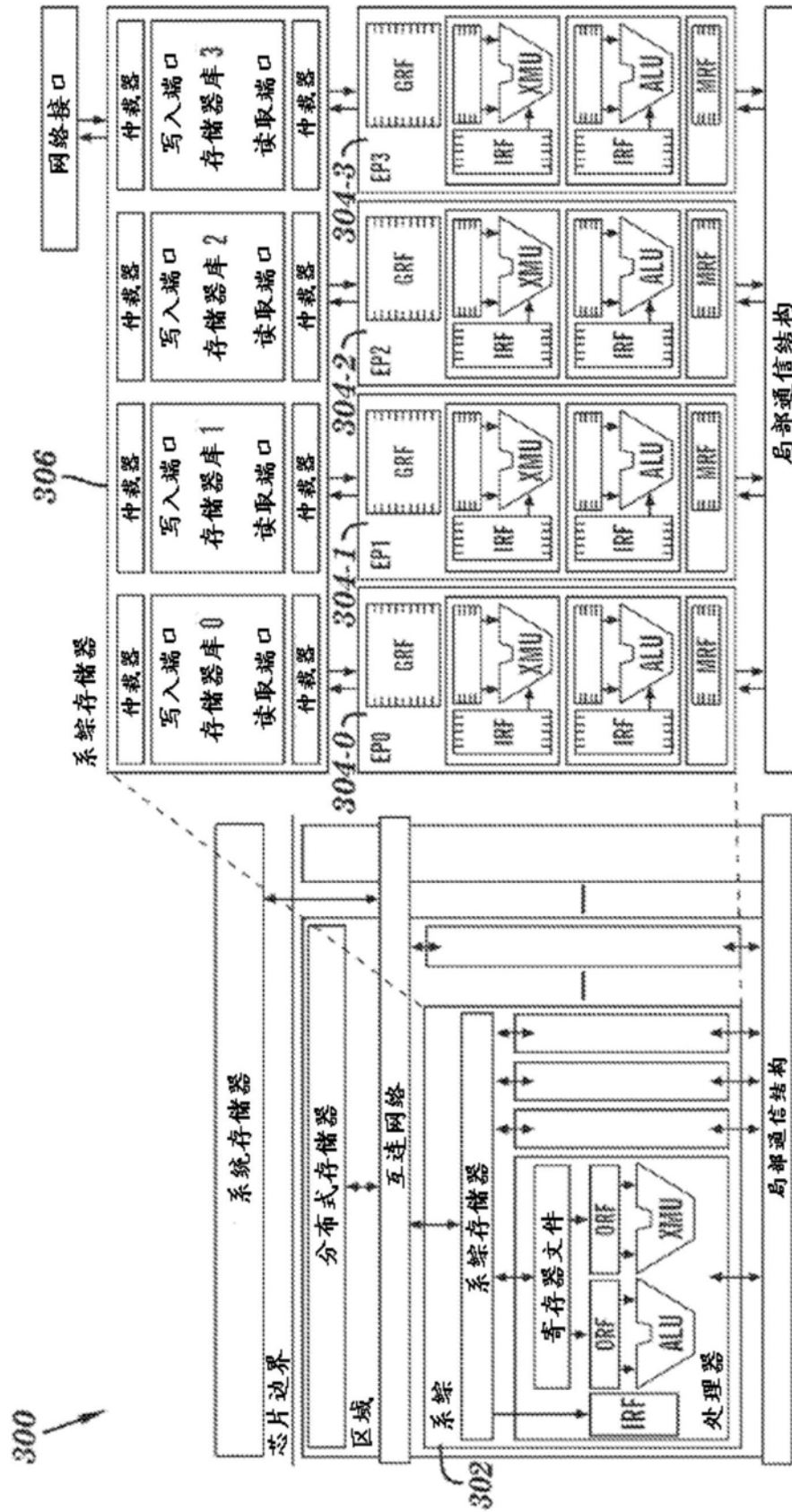


图3

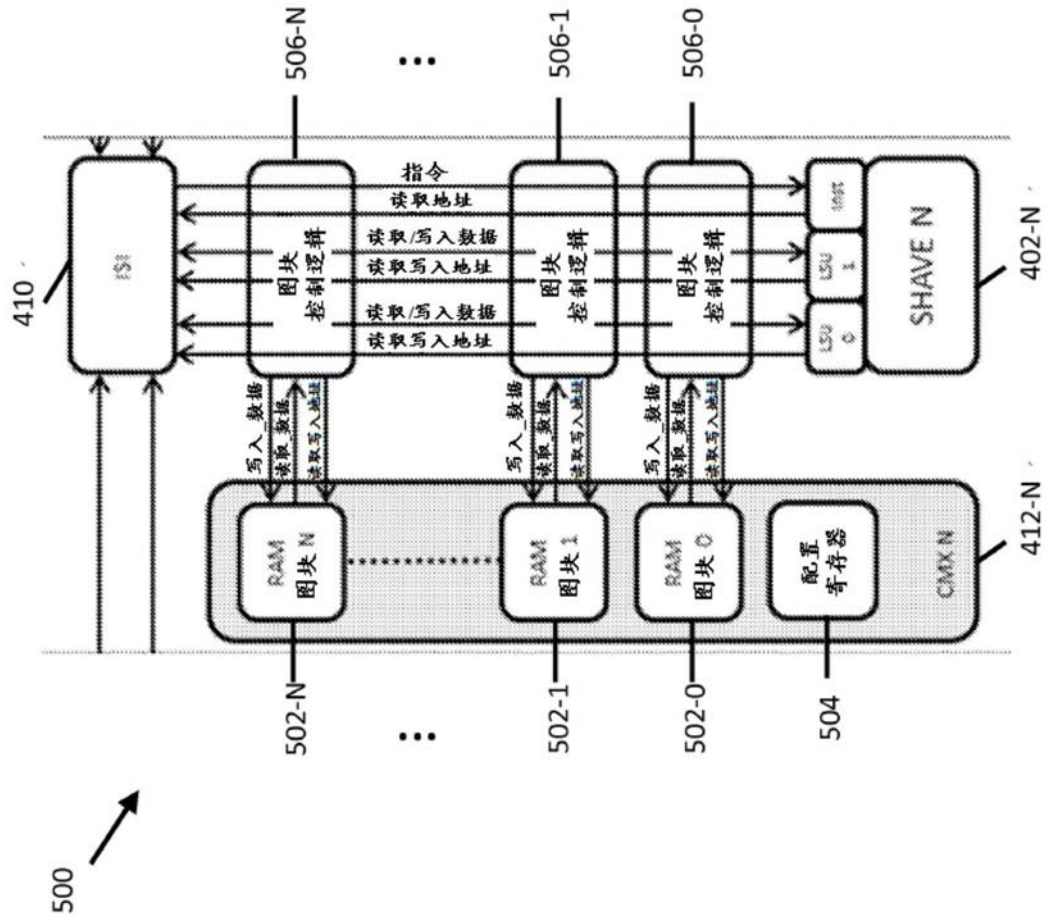


图5

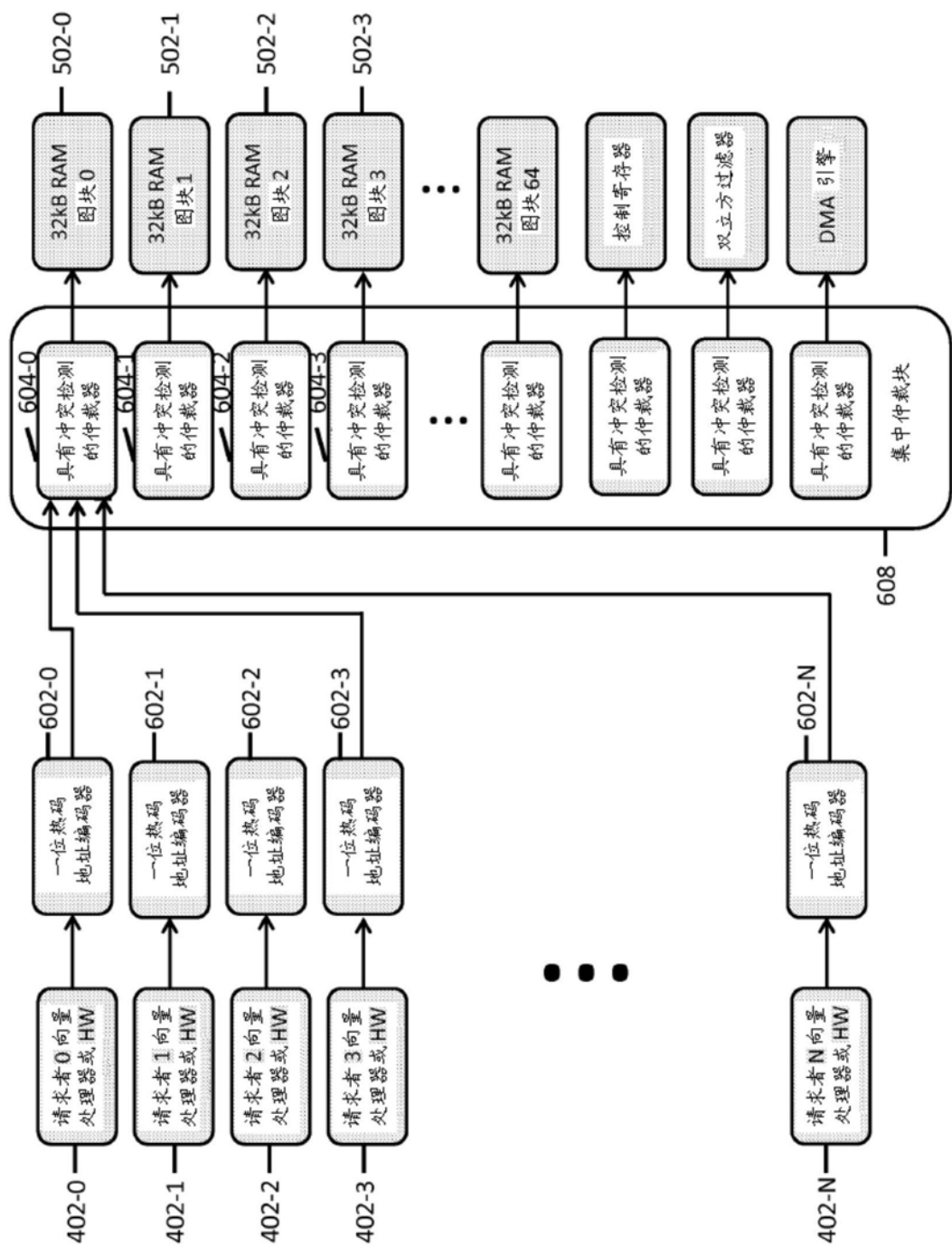


图6

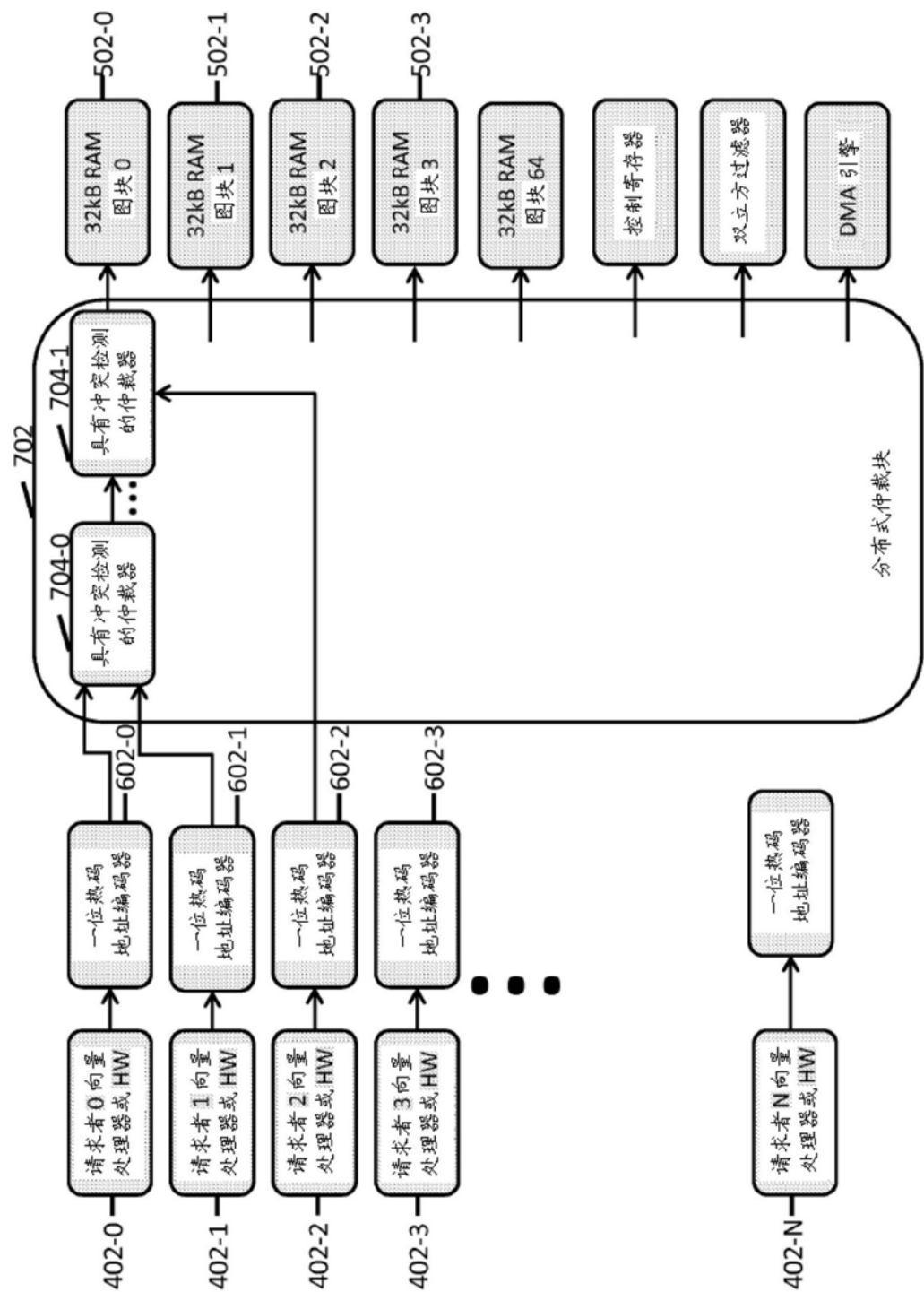


图7

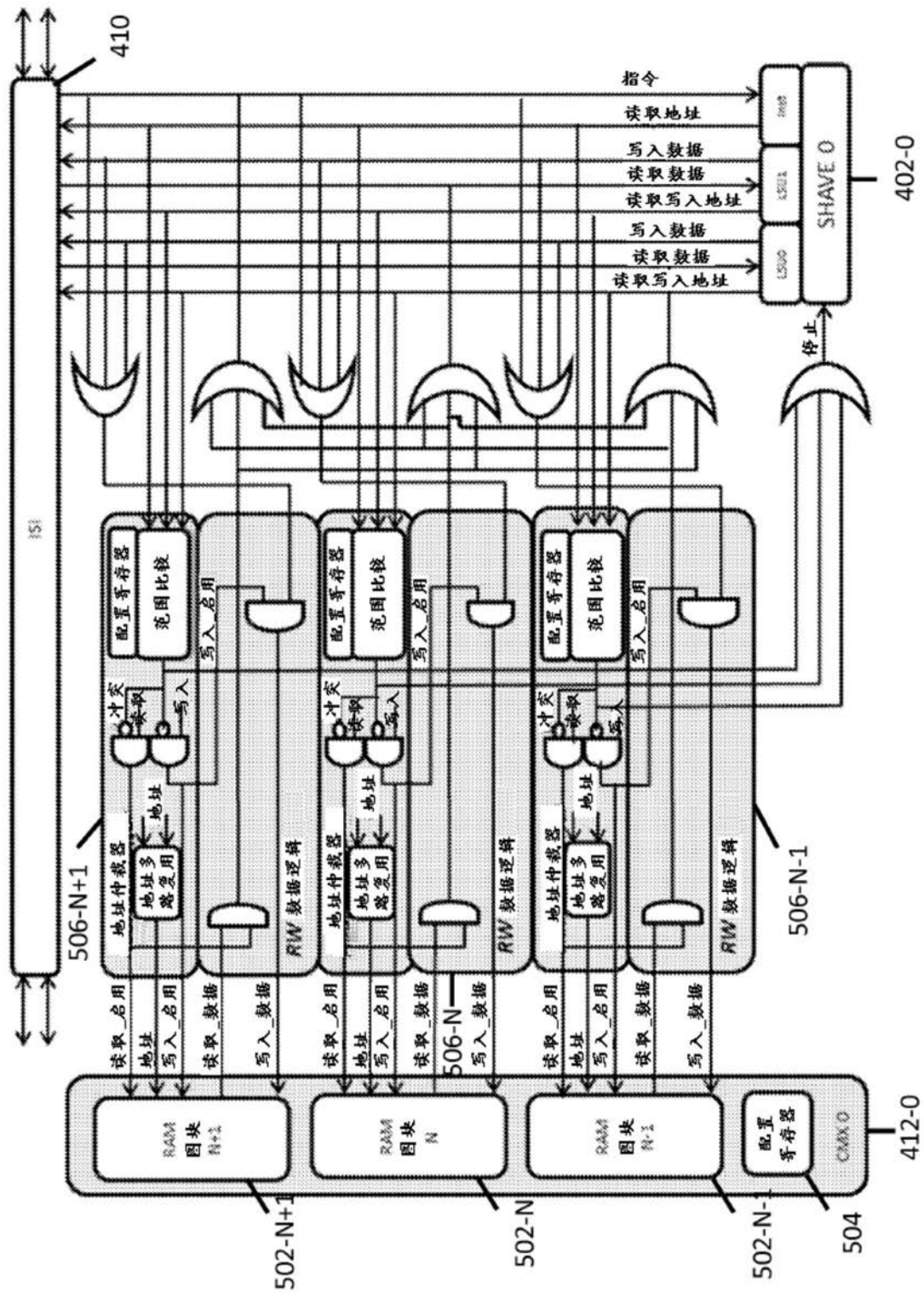


图8

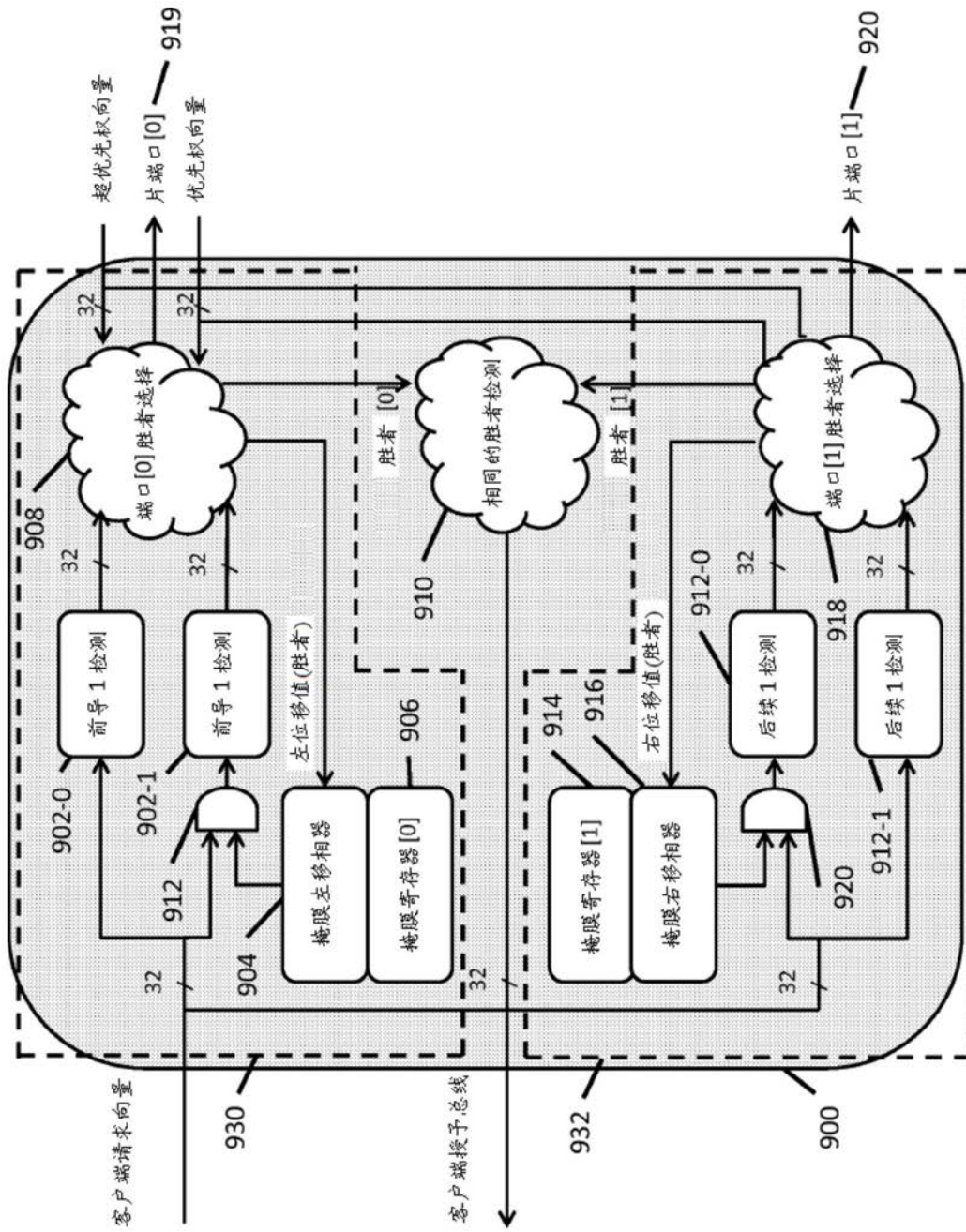


图9

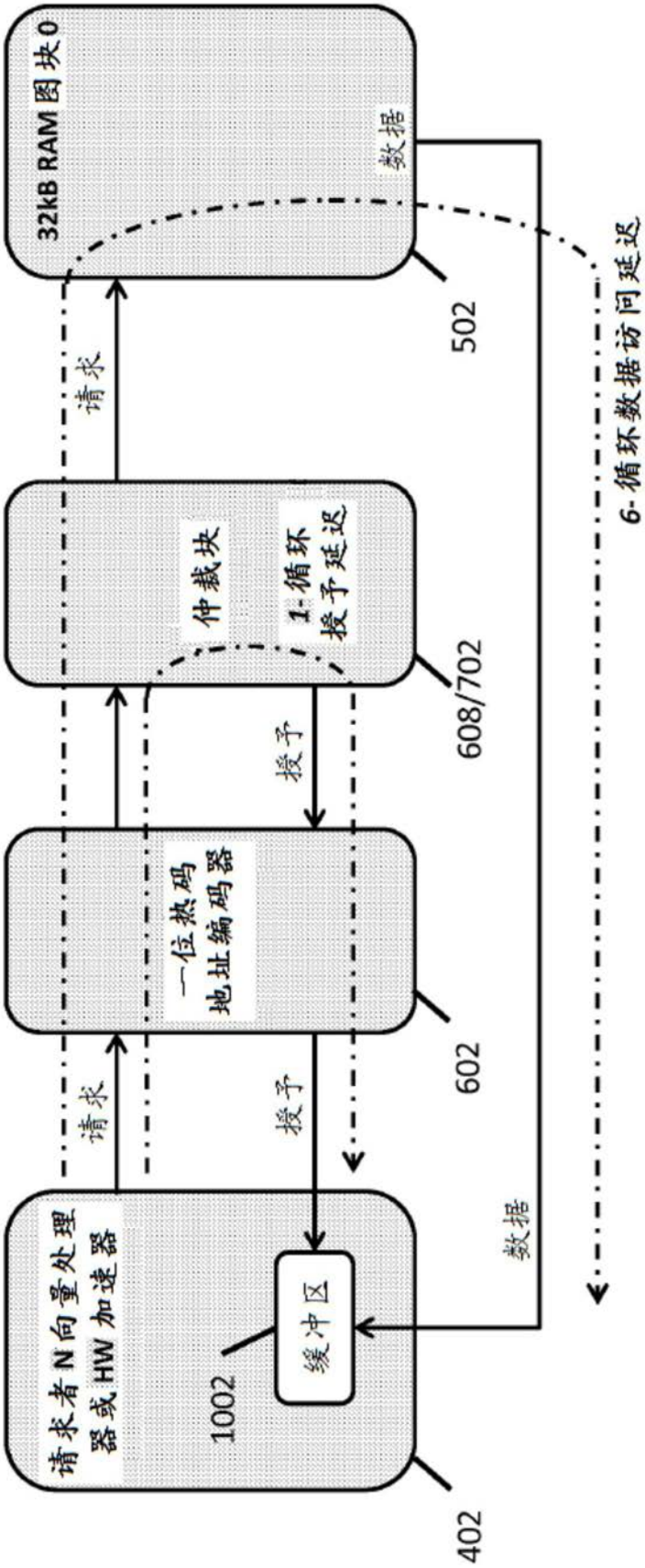


图10

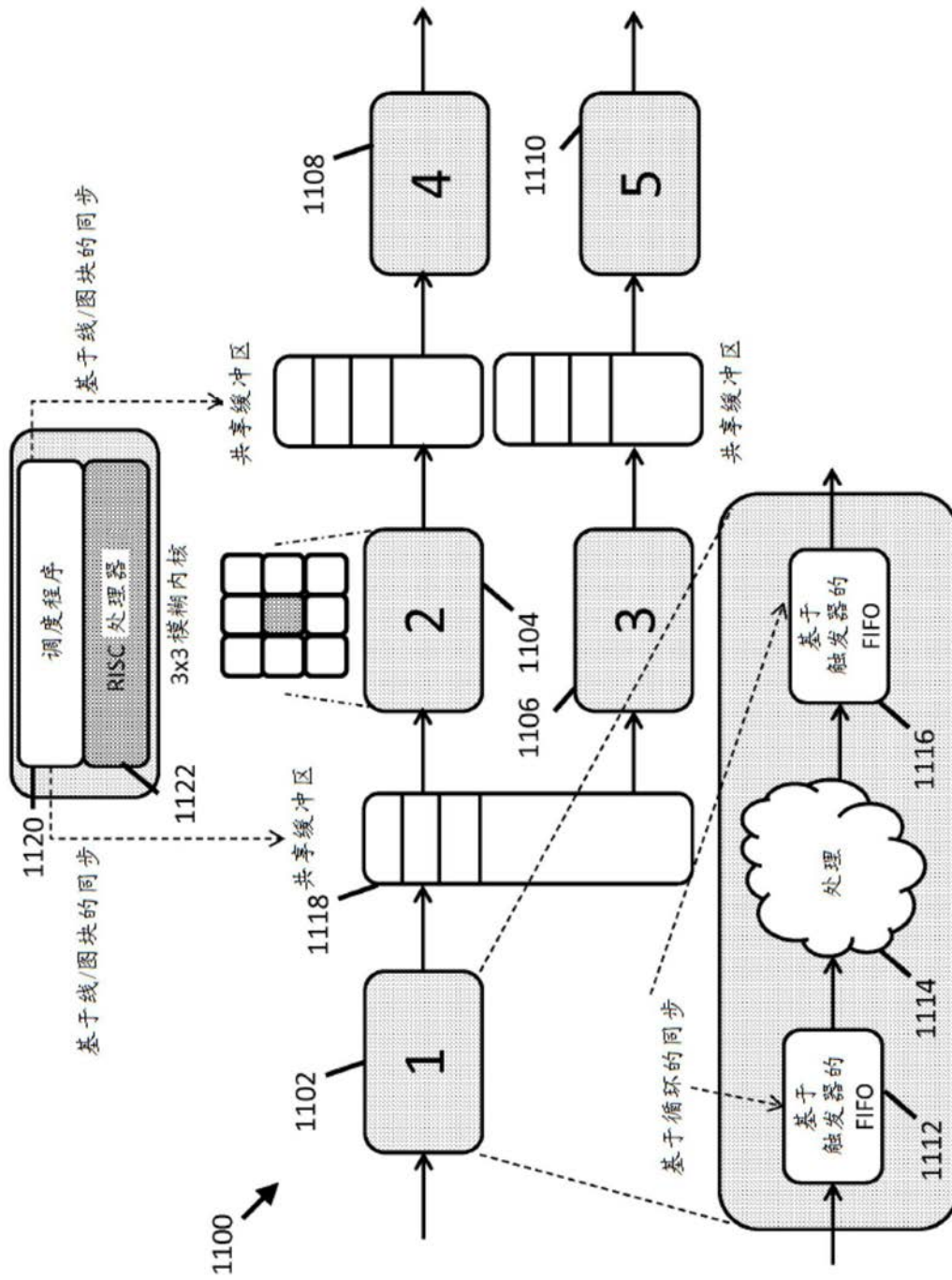


图11

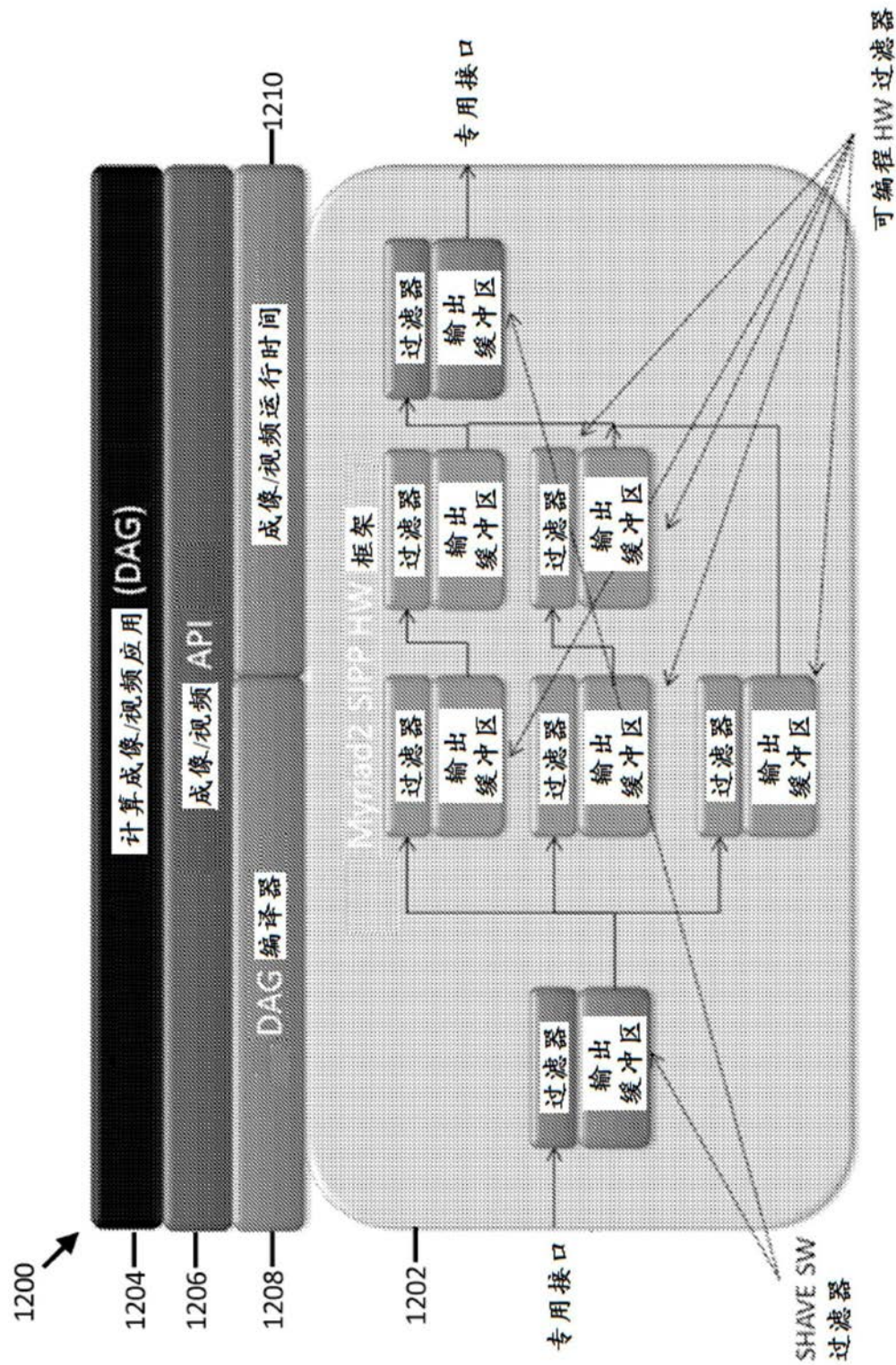


图12

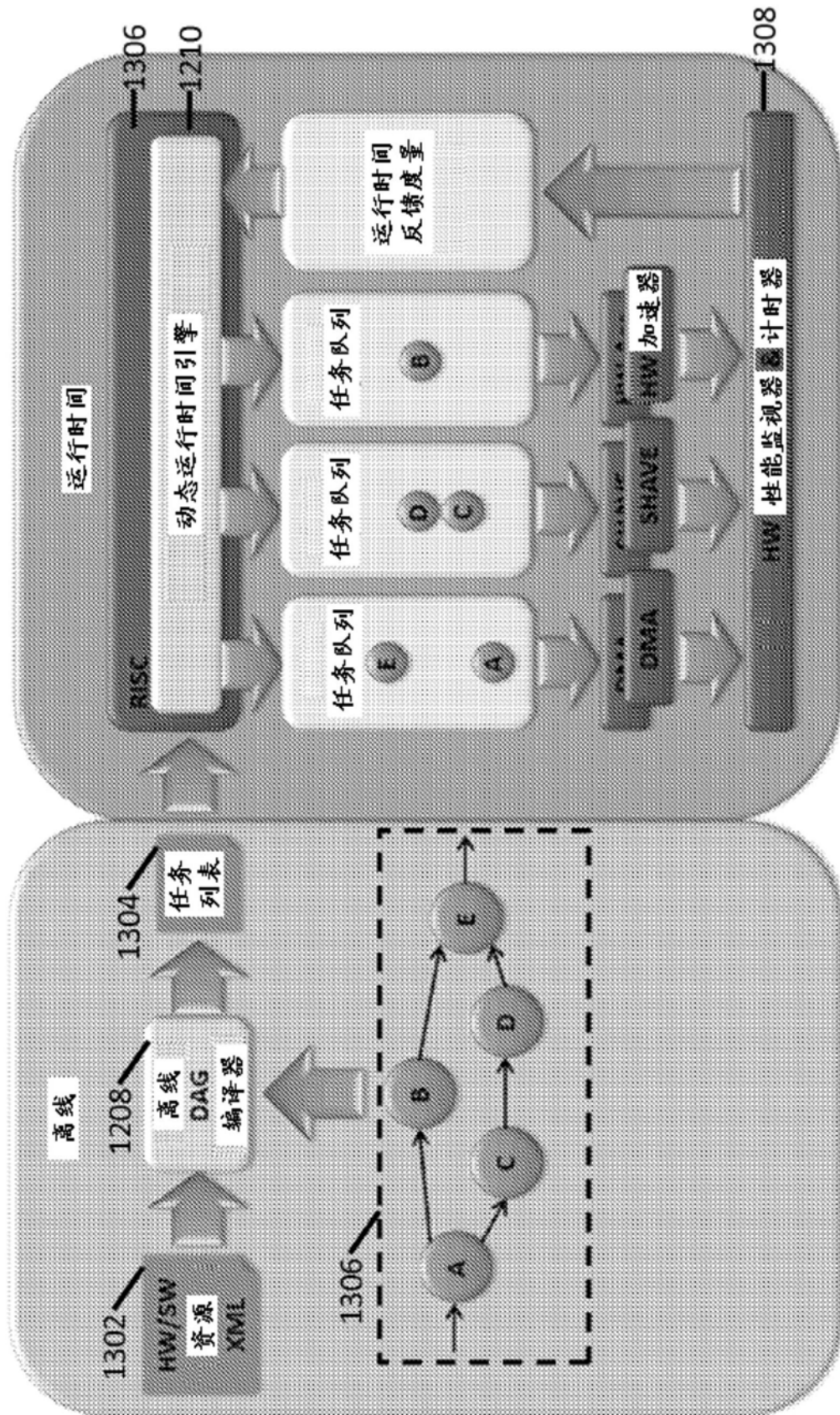


图13

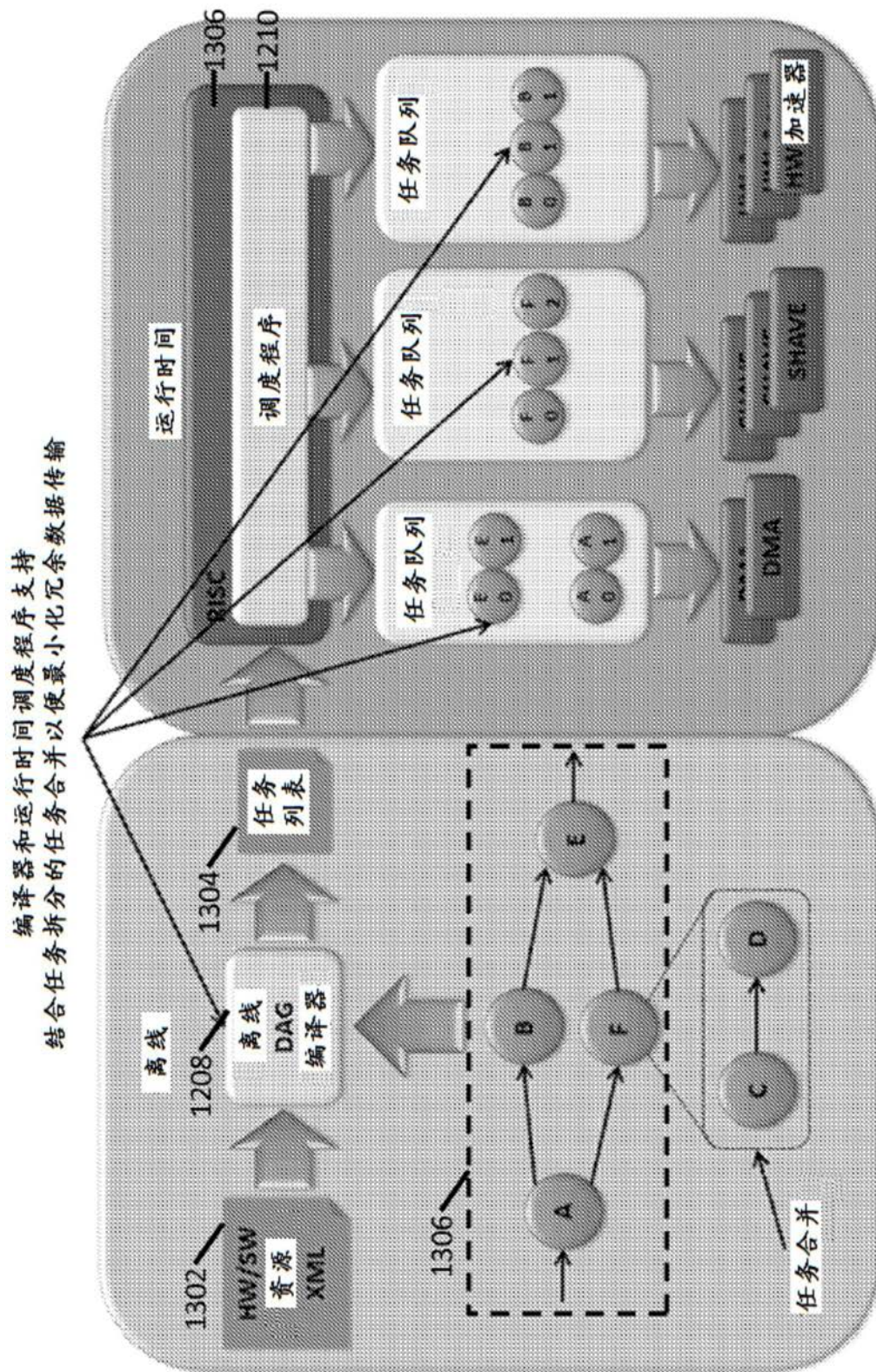


图14A

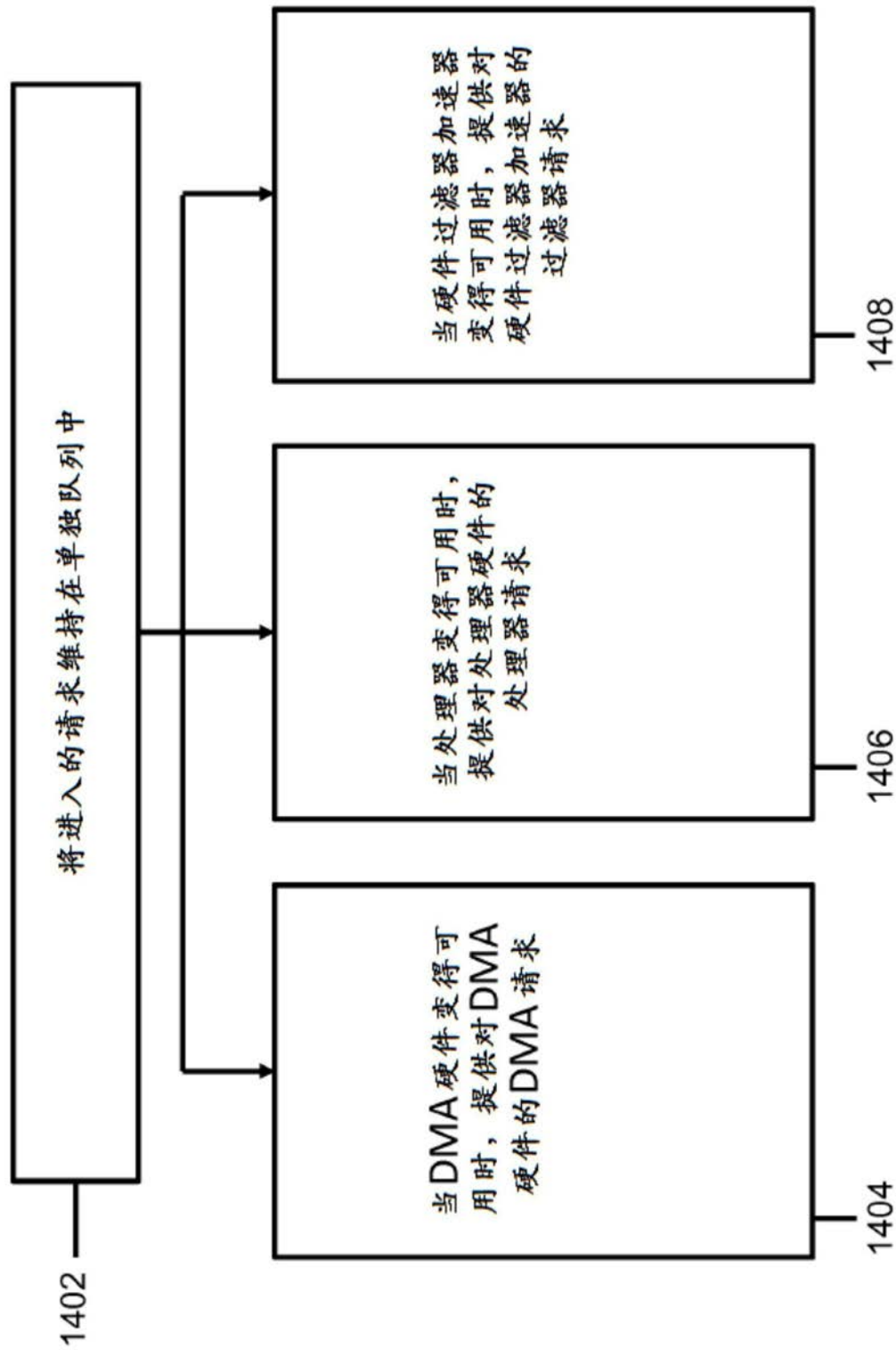


图14B

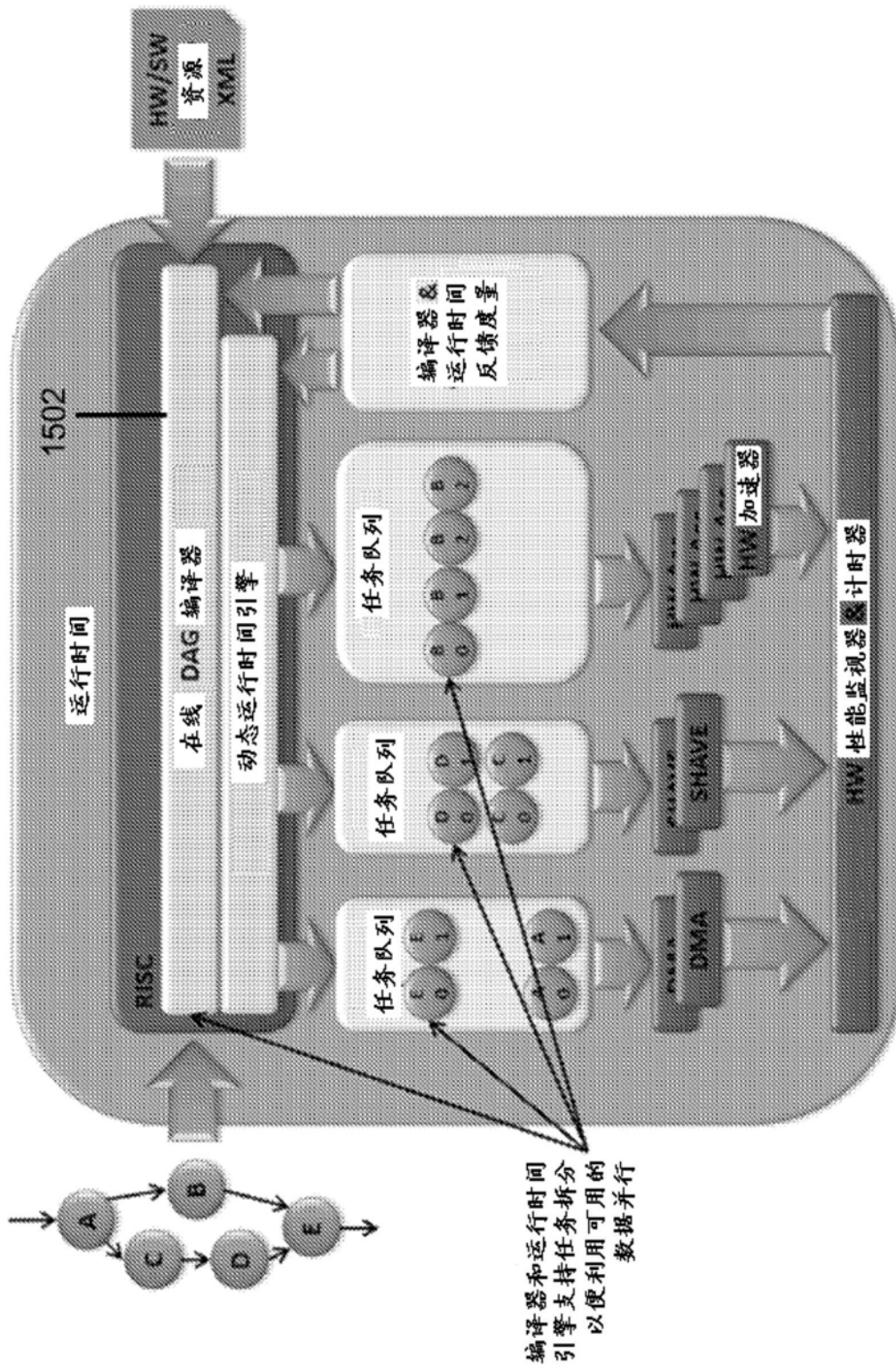


图15

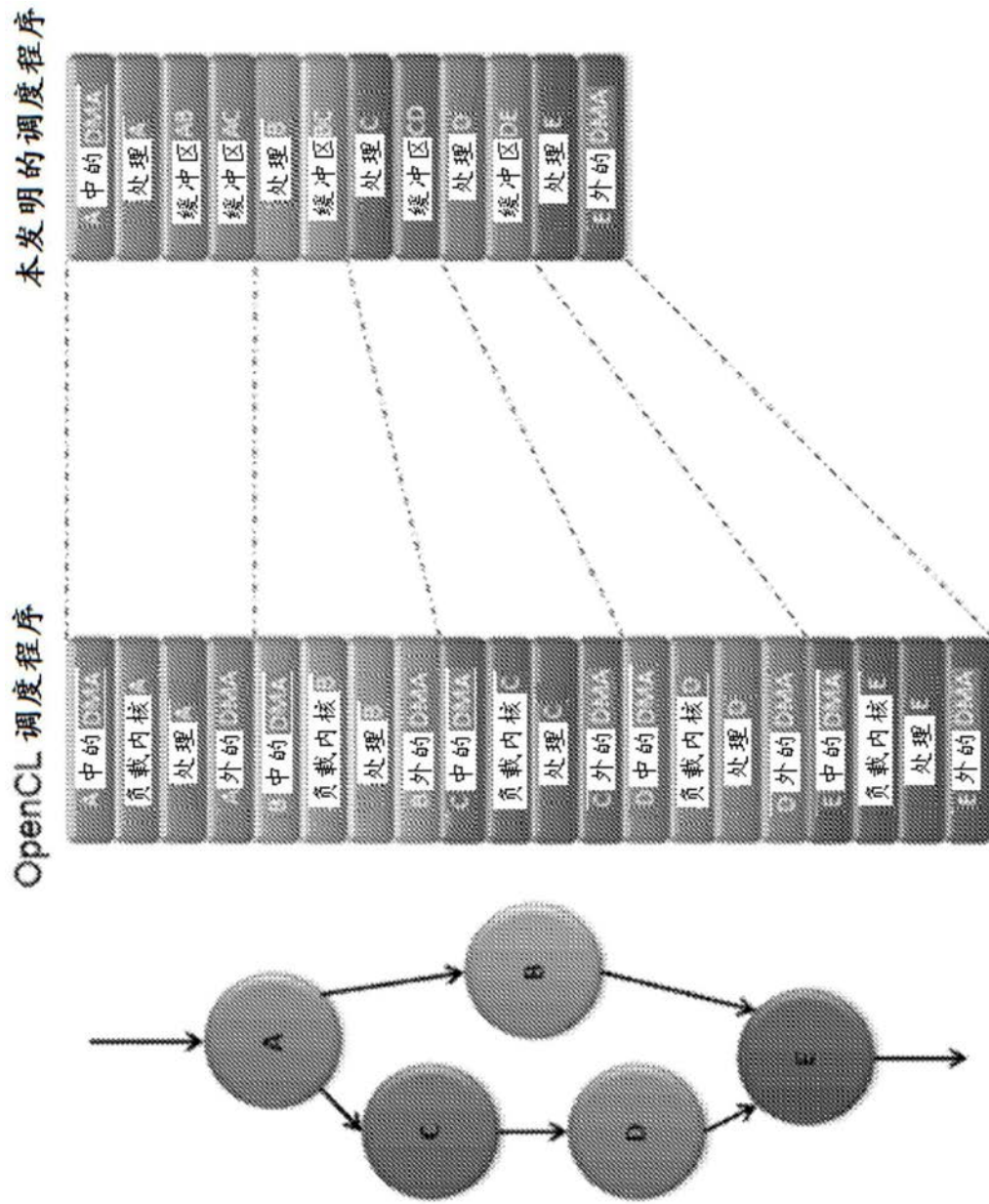


图16

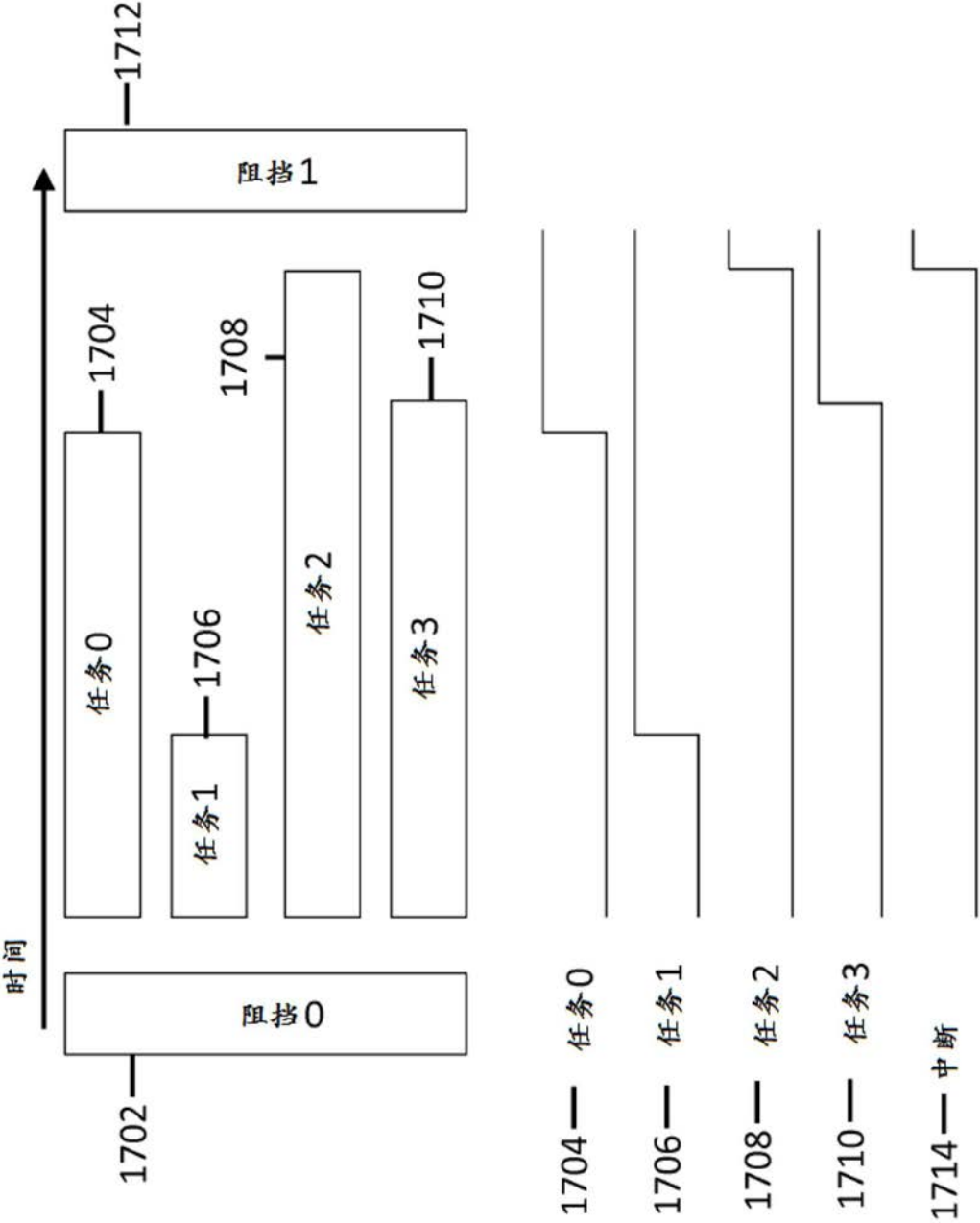


图17

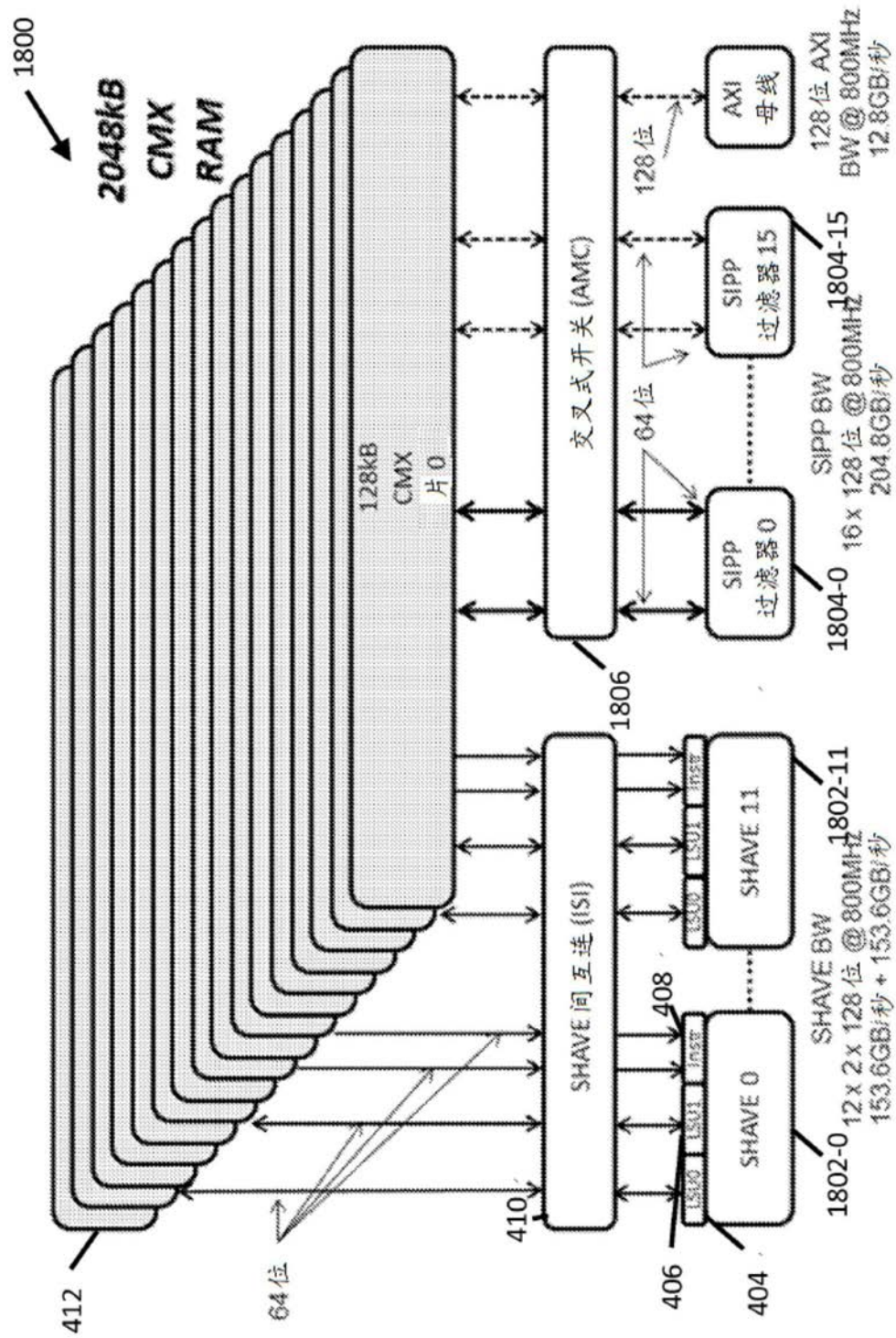


图18

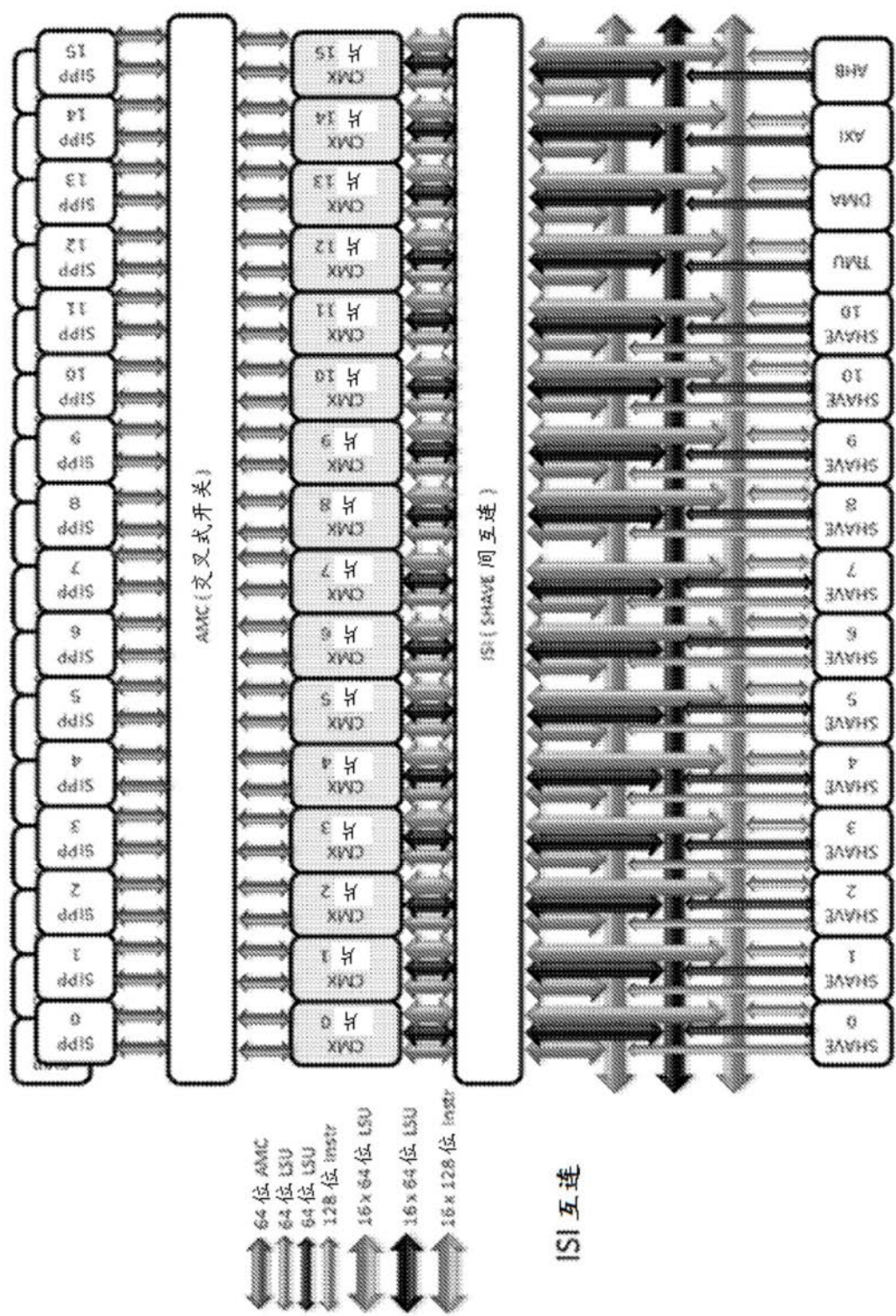


图19

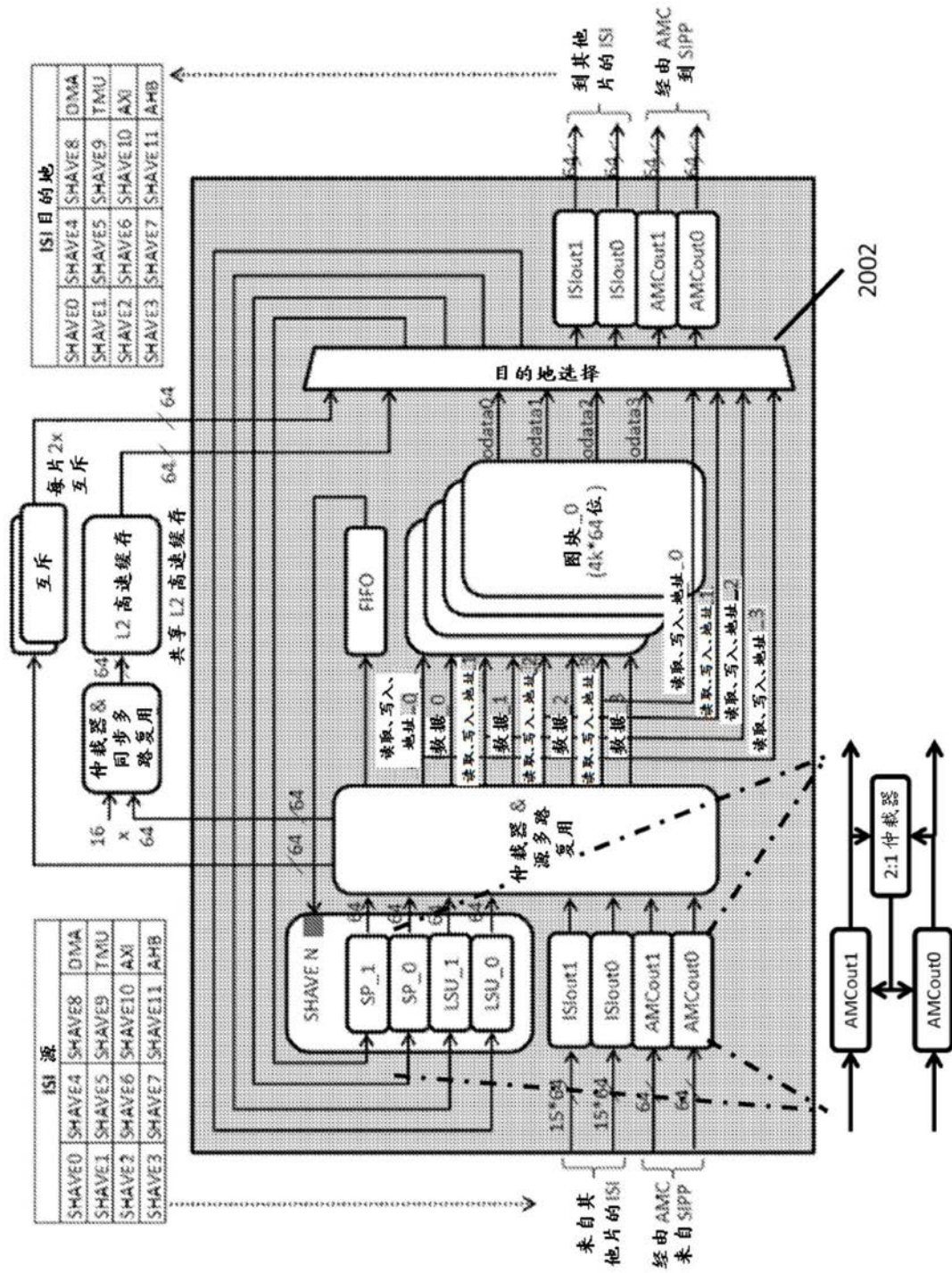


图20

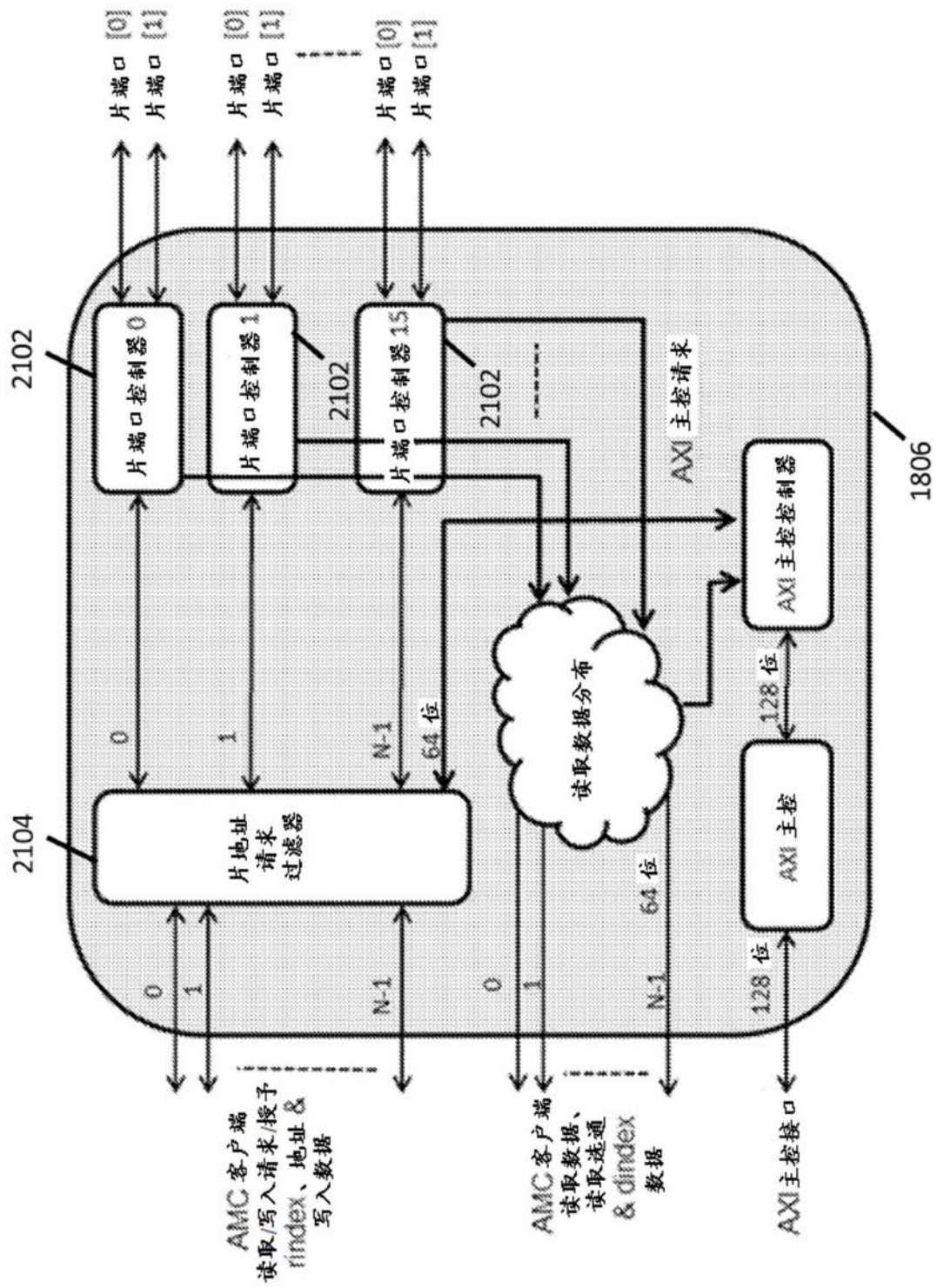


图21

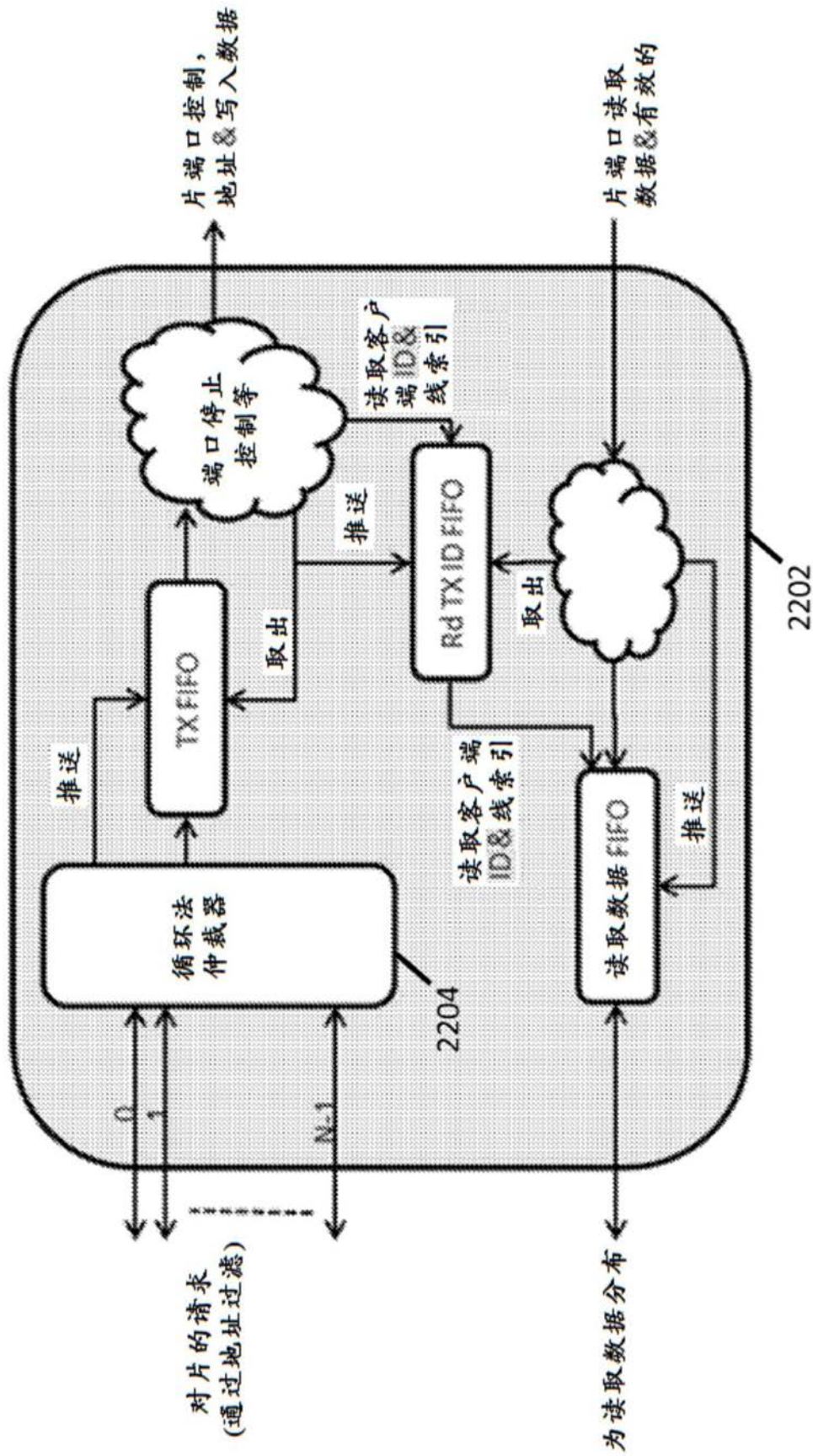


图22

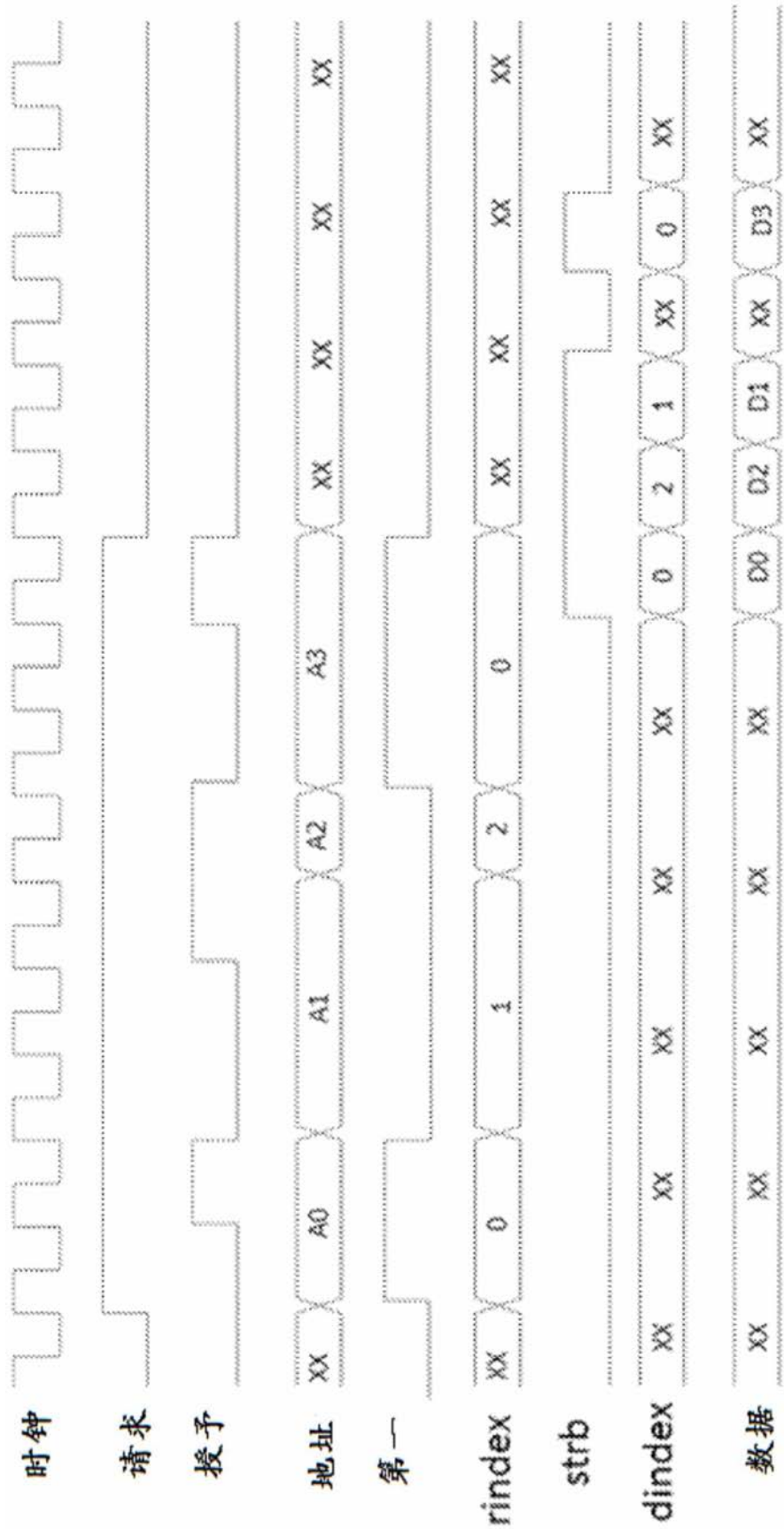


图23

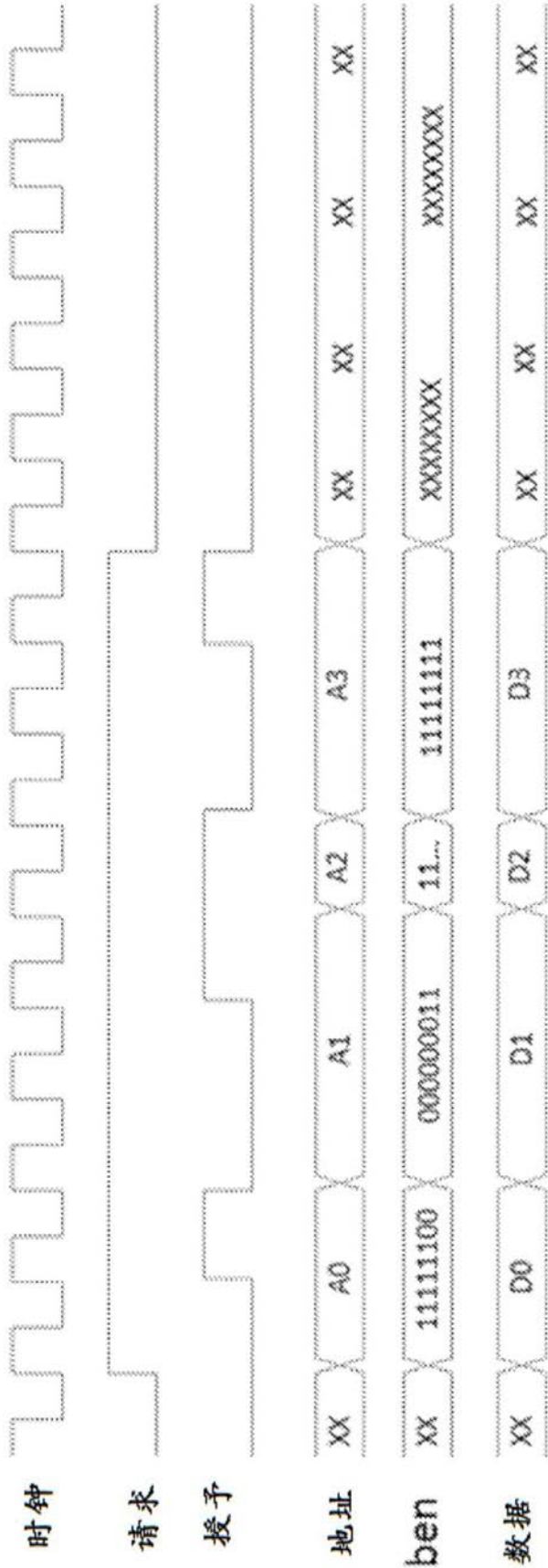


图24

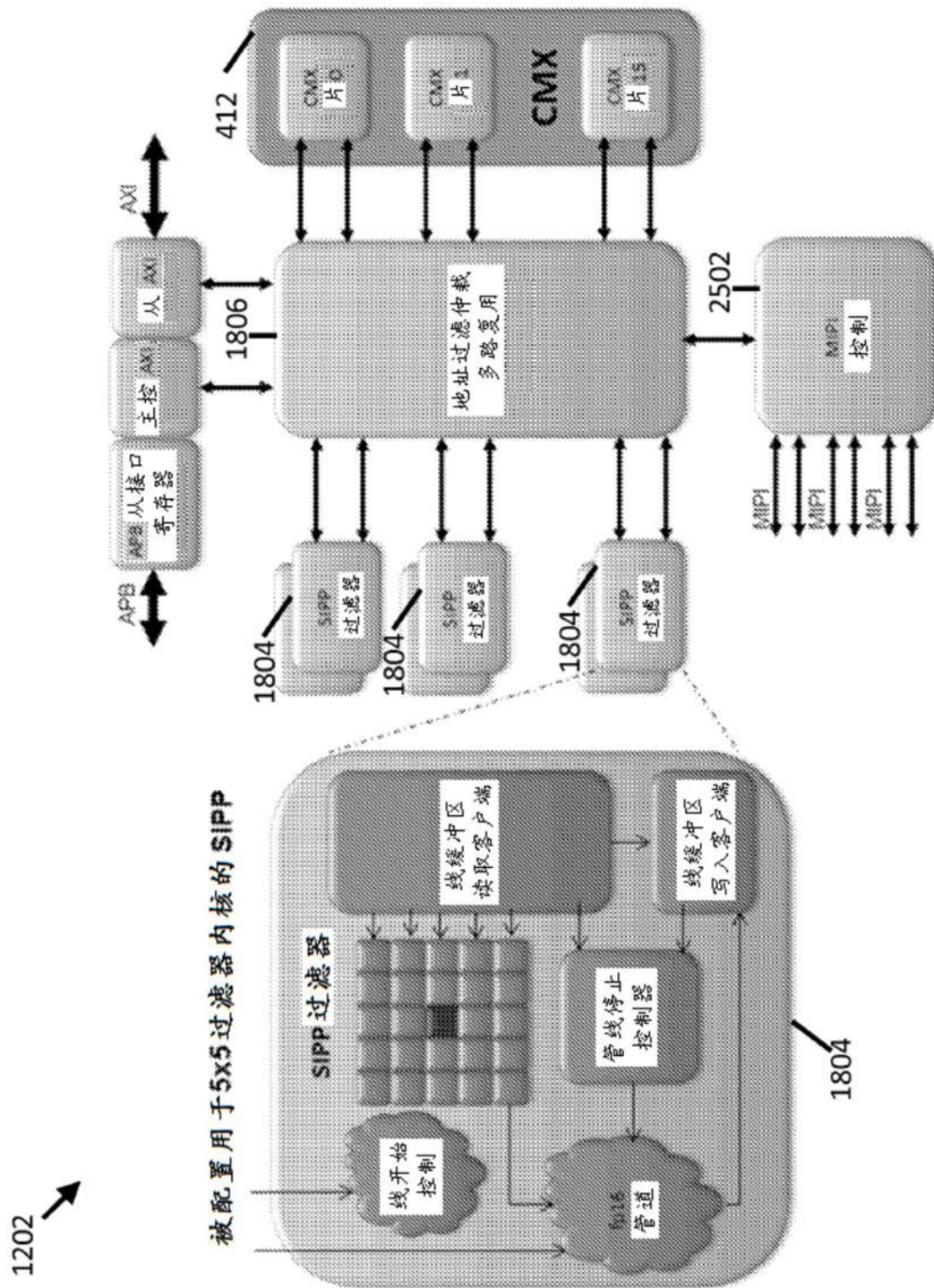


图25

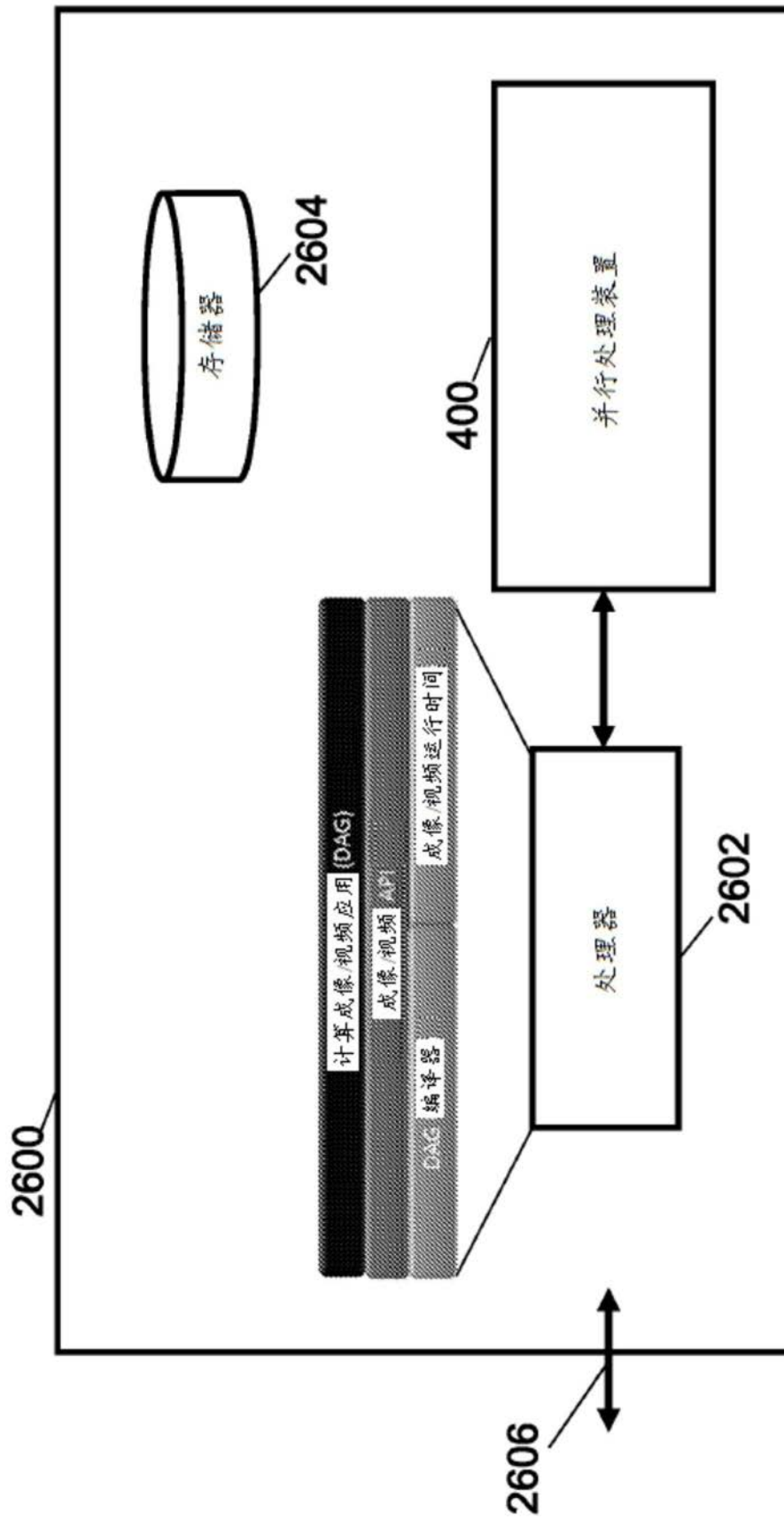


图26