(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102375800 A
(43)申请公布日 2012.03.14

(21)申请号 201110160959.8

(22)申请日 2011.06.09

(30)优先权数据
61/372,563 2010.08.11 US
13/074,034 2011.03.29 US

(71)申请人 普莱姆森斯有限公司
地址 以色列特拉维夫市

(72)发明人 I·萨尔

(74)专利代理机构 北京北翔知识产权代理有限公司 11285
代理人 徐燕 杨勇

(51)Int.Cl.
G06F 15/80(2006.01)

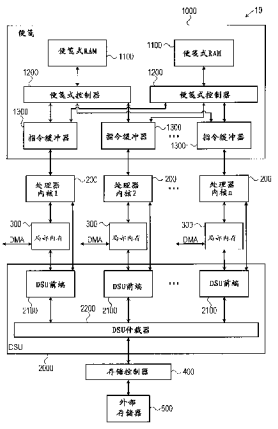
权利要求书 2 页 说明书 11 页 附图 7 页

(54)发明名称

用于机器视觉算法的多处理器片上系统

(57)摘要

一种多处理器系统，包括主存储器和多个处理核，该多个处理核被配置为执行使用存储在该主存储器中的数据的软件。在一些实施方案中，该多处理器系统包括数据流单元，其连接在该多个处理核和该主存储器之间，并被配置为从该主存储器中预取得数据，以供该多个处理核使用。在一些实施方案中，该多处理器包括便笺式处理单元，其连接至该多个处理核并被配置，以为该多个处理核执行所述软件的选定部分，该选定部分使得所述多个处理核中的两个或更多个处理核并发地访问给定的数据项目。



CN 102375800 A

1. 一种多处理器系统，包括：

主存储器；

多个处理核，被配置为执行使用存储在所述主存储器中的数据的软件；

数据流单元，连接在所述多个处理核和所述主存储器之间，并被配置为从所述主存储器中预取得所述数据，以供所述多个处理核使用。

2. 根据权利要求 1 所述的多处理器系统，其中所述数据流单元被配置，从而为所述多个处理核将数据存储在所述主存储器中。

3. 根据权利要求 1 所述的多处理器系统，其中所述数据流单元包括仲裁电路，该仲裁电路被配置，从而为所述多个处理核中的两个或更多个处理核决定对所述主存储器进行的同时访问。

4. 根据权利要求 1 所述的多处理器系统，其中所述数据流单元对于每一处理核包括一个相应的前端单元，该前端单元被配置为从该处理核接收所述主存储器中的相应的地址列表，以及根据该列表从所述主存储器中预取得所述数据。

5. 根据权利要求 4 所述的多处理器系统，还包括与每一处理核相关联的相应的局部内存，其中该每一处理核和其对应的前端单元被配置为经由该相应的局部内存来交换数据。

6. 根据权利要求 5 所述的多处理器系统，其中所述每一处理核和其对应的前端单元被配置为将所述地址列表维持在循环缓冲器中，所述循环缓冲器被存储在相应的局部内存中。

7. 根据权利要求 1 所述的多处理器系统，其中至少所述多个处理核和所述数据流单元被包括在单个集成电路中。

8. 一种多处理器系统，包括：

主存储器；

多个处理核，被配置为执行使用存储在所述主存储器中的数据的软件；

便笺式处理单元，连接至所述多个处理核并被配置，从而为所述多个处理核执行所述软件的选定部分，该选定部分使得所述多个处理核中的两个或更多个处理核并发地访问给定的数据项目。

9. 根据权利要求 8 所述的多处理器系统，其中所述便笺式处理单元包括专用存储器，该专用存储器用于存储由所述两个或更多个处理核访问的给定的数据项目。

10. 根据权利要求 9 所述的多处理器系统，其中所述便笺式处理单元被配置为从所述多个处理核接受便笺式指令，仲裁所述便笺式指令，以及在所述专用存储器中执行所仲裁的便笺式指令。

11. 根据权利要求 8 所述的多处理器系统，其中至少所述多个处理核和所述便笺式处理单元被包括在单个集成电路中。

12. 一种用于数据处理的方法，包括：

在多处理器系统的多个处理核中执行使用存储在主存储器中的数据的软件；以及

通过连接在所述多个处理核和所述主存储器之间的数据流单元，从所述主存储器预取得所述数据，以供所述多个处理核使用。

13. 根据权利要求 12 所述的方法，还包括，通过所述数据流单元为所述多个处理核将数据存储在所述主存储器中。

14. 根据权利要求 12 所述的方法，其中预取得所述数据包括，为所述多个处理核中的两个或更多个处理核决定对所述主存储器进行的同时访问。

15. 根据权利要求 12 所述的方法，其中预取得所述数据包括，从每一处理核向相应的前端单元提供所述主存储器中的相应的地址列表，以及根据所述列表通过所述前端单元从所述主存储器中预取得所述数据。

16. 根据权利要求 15 所述的方法，其中预取得所述数据包括，经由与每一处理核关联的相应的局部内存，在该每一处理核和相应的前端单元之间交换所述数据。

17. 根据权利要求 16 所述的方法，其中交换所述数据包括，将所述地址列表维持在相应的循环缓冲器中，所述相应的循环缓冲器存储在相应的局部内存中。

18. 根据权利要求 12 所述的方法，其中至少所述多个处理核和所述数据流单元被包括在单个集成电路中。

19. 一种用于数据处理的方法，包括 ：

在多处理器系统的多个处理核中执行使用存储在主存储器中的数据的软件 ；以及

使用便笺式处理单元，所述便笺式处理单元连接至所述多个处理核，为所述多个处理核执行所述软件的选定部分，所述选定部分使得所述多个处理核中的两个或更多个处理核并发地访问给定的数据项目。

20. 根据权利要求 19 所述的方法，其中执行所述软件的选定部分包括，将由所述两个或更多个处理核访问的所述给定的数据项目存储在所述便笺式处理单元的专用存储器中。

21. 根据权利要求 20 所述的方法，还包括，在所述便笺式处理单元中从所述多个处理核接受便笺式指令，仲裁所述便笺式指令，以及在所述专用存储器中执行所仲裁的便笺式指令。

22. 根据权利要求 19 所述的方法，其中至少所述多个处理核和所述便笺式处理单元被包括在单个集成电路中。

用于机器视觉算法的多处理器片上系统

技术领域

[0001] 本发明大体涉及多处理器系统，并更具体而言涉及用于在多处理器系统中有效利用共享资源的方法和系统。

背景技术

[0002] 近年来，随着 VLSI 器件的密度逐渐增大和越来越复杂的运算任务——诸如实时机器视觉所需的那些运算任务的出现，完全多处理器系统 (complete multiprocessor system)——特别是单个单片 (monolithic) 器件中的对称多处理 (SMP) 系统——变得流行起来。在一些多处理器系统中，存储器资源由多个多处理器共享。然而，这一共享可能产生存储一致性 (memory coherency) 问题，并造成瓶颈。

[0003] 在美国专利 7,529,799（其公开内容以引证方式纳入本说明书）中，发明者们介绍了一种大型 SMP 系统的分布式系统结构，该结构使用基于总线的高速缓存一致性协议 (cache-coherence protocol)。所述分布式系统结构包括地址切换、多个存储器子系统以及多个主设备 (master device)，或者是处理器、I/O 代理或者是一致性存储适配器，它们被组织成一组由节点控制器支持的节点。所述节点控制器从主设备接收事务 (transaction)，作为另一主设备或作为从设备与主设备通信，并为从主设备接收的事务排队。由于一致性的实现在时间上和空间上是分布式的，所以所述节点控制器帮助维持了高速缓存一致性。此外，用于标准总线协议的事务标签格式被扩展，以确保在整个系统中维持唯一的事务标签。边带信号被用来进行干预以及再运行，以在某些情况下将事务标签保留在节点控制器处。

[0004] 在美国专利 7,237,071（其公开内容以引证方式纳入本说明书）中，介绍了一种 SMP 系统，该系统具有由相同的处理器组成的并行多处理体系结构，并包括单个程序存储器。程序访问仲裁逻辑向单个请求中央处理单元一次提供一个指令。共享存储器访问仲裁逻辑可以从分立的可同时访问的存储器组提供数据，或在中央处理单元之间进行访问仲裁。该系统可以通过以下方式模拟原子读取 / 修改 / 写入指令 :在对所述共享存储器中的一组预定地址之一进行读取访问之后的一预定数量的存储周期内，禁止另一中央处理单元访问这一地址。

发明内容

[0005] 本发明的一个实施方案提供了一种多处理器系统，该多处理器系统包括主存储器、多个处理核以及数据流单元 (data streaming unit)。所述多个处理核被配置为执行使用存储在所述主存储器中的数据的软件。所述数据流单元，连接在所述多个处理核和所述主存储器之间，并被配置为从所述主存储器中预取得数据，以供所述多个处理核使用。

[0006] 在一些实施方案中，所述数据流单元被配置，从而为 (on behalf of) 所述多个处理核将数据存储在所述主存储器中。在一个实施方案中，所述数据流单元包括仲裁电路，该仲裁电路被配置，从而为所述多个处理核中的两个或更多个处理核决定 (resolve) 对所述

主存储器进行的同时访问。

[0007] 在一些实施方案中，所述数据流单元对于每一处理核包括一个相应的前端单元，该前端单元被配置为从该处理核接收所述主存储器中的相应的地址列表，以及根据所述列表从所述主存储器中预取得所述数据。在本文公开的一个实施方案中，所述多处理器系统包括与每一处理核相关联的相应的局部内存，其中每一处理核和其对应的前端单元被配置为经由该相应的局部内存来交换数据。

[0008] 在一个实施方案中，每一处理核和其对应的前端单元被配置，从而将所述地址列表维持在循环缓冲器(circular buffer)中，所述循环缓冲器被存储在相应的局部内存中。在一些实施方案中，至少所述多个处理核和所述数据流单元被包括在单个集成电路中。

[0009] 根据本发明的一个实施方案，还提供了一种多处理器系统，该多处理器系统包括主存储器、多个处理核以及便笺式处理单元(scratch-padprocessing unit)。所述多个处理核被配置为执行使用存储在所述主存储器中的数据的软件。所述便笺式处理单元连接至所述多个处理核并被配置，从而为所述多个处理核执行所述软件的选定部分，所述选定部分使得所述多个处理核中的两个或更多个处理核并发地访问给定的数据项目。

[0010] 在一些实施方案中，所述便笺式处理单元包括专用存储器，该专用存储器用于存储由所述两个或更多个处理核访问的给定的数据项目。在一个实施方案中，所述便笺式处理单元被配置为从所述多个处理核接受便笺式指令，仲裁所述便笺式指令，以及在所述专用存储器中执行所仲裁的便笺式指令。在本文公开的一个实施方案中，至少所述多个处理核和所述便笺式处理单元被包括在单个集成电路中。

[0011] 根据本发明的一个实施方案，还提供了一种用于数据处理的方法。该方法包括在多处理器系统的多个处理核中执行使用存储在主存储器中的数据的软件。通过连接在所述多个处理核和所述主存储器之间的数据流单元，从所述主存储器预取得数据，以供多个处理核使用。

[0012] 根据本发明的一个实施方案，还提供了一种用于数据处理的方法。该方法包括，在多处理器系统的多个处理核中执行使用存储在主存储器中的数据的软件。使用连接至所述多个处理核的便笺式处理单元，为所述多个处理核执行所述软件的选定部分，所述选定部分使得所述多个处理核中的两个或更多个处理核并发地访问给定的数据项目。

[0013] 结合附图，将从下文对实施方案的详细描述中得到对本发明的更充分理解。

附图说明

[0014] 图 1 是根据本发明的一个实施方案的框图，其示意性示出了一个多处理器系统。

[0015] 图 2 是根据本发明的一个实施方案的框图，其示意性地示出了一个数据流单元(DSU)

[0016] 图 3 是根据本发明的一个实施方案的图示，其示意性地示出了一个循环缓冲器的结构。

[0017] 图 4 是根据本发明的一个实施方案的框图，其示意性地示出了一个 DSU 前端的结构。

[0018] 图 5 是根据本发明的一个实施方案的框图，其示意性地示出了该 DSU 前端中的一个缓冲器管理单元的结构。

[0019] 图 6 是根据本发明的一个实施方案的框图，其示意性地示出了一个 DSU 仲裁器的结构。

[0020] 图 7 是根据本发明的一个实施方案的框图，其示意性地示出了一个便笺式单元的结构及其所连接的系统元件。

[0021] 图 8 是根据本发明的一个实施方案的框图，其示意性地示出了一个便笺式控制器。

具体实施方式

[0022] 概况

[0023] 一些多处理器系统被实现在单个集成电路（片上系统，或 SOC）中。该 SOC 通常包括局部内存单元的一个或多个实例（instance），但是并不包括主存储器——其可以远大于所述局部内存。所述主存储器通常被实现在一个或多个集成电路中，允许满足顺序（猝发（burst））访问的高带宽但具有长的等待时间。当这样的主存储器在多处理器系统中被多个处理器共享时，应实施有效率的仲裁，以避免因排队访问存储器而引起的严重的性能下降。

[0024] 本发明的实施方案介绍了一种新方法，以减轻多处理器系统中由多个处理器访问共享存储器资源所引起的性能瓶颈。根据本发明的一些实施方案，该多处理器系统包括数据流单元（DSU），该数据流单元在处理器内核需要数据之前就从主存储器取得该数据。该 DSU 将所取得的数据存储在连接至处理器内核的局部内存中，在该局部内存中这些数据可以在需要时被处理器内核访问。该 DSU 也将数据从局部内存写入主存储器。

[0025] 与多处理器系统相关联的另一问题是，当两个或更多个处理器访问相同的存储位置时确保存储器一致性（memory coherency）。在一些由多处理器系统执行的图像处理定向算法中，很少出现多个处理器内核访问主存储器中的同一地址的情形，而这种情形仅发生在例如这样的任务中：从图像的由多个处理器内核处理的各区域中收集图像统计数据。然而，如果未得到有效率的应对，即使这样的情形很少发生，却仍然会产生性能瓶颈。本发明的实施方案引入了减轻这一瓶颈的一种新方法。根据本文公开的实施方案，对共享存储位置的所有访问均由便笺式单元操控，该便笺式单元包括专用处理器和小型局部内存，并执行这样的软件，该软件为执行访问共享存储位置的并行任务而被优化。该便笺式单元通常作为协处理器（co-processor）而附接至该多处理器系统的处理器内核。

[0026] 系统描述

[0027] 图 1 是根据本发明的一个实施方案的框图，其示意性地示出了多处理器系统 10。多处理器系统 10 的所有所示出的元件可以位于单个集成电路中，并构成一个片上系统（SOC）。

[0028] 多处理器系统 10 包括主存储器 500（也被称为外部存储器）。在一些实施方案中，主存储器 500 包括一个或多个分立的集成电路，且不是该多处理器 SOC 的一部分。在另一实施方案中，主存储器和多处理器系统 10 的其他元件位于同一 SOC 中。在另一实施方案中，主存储器 500 可以包括多个部分，其中一些位于该 SOC 内，另一些位于一个或多个外部芯片中。在接下来的描述中，术语“外部存储器”将被用于主存储器；不过本发明决不限于未在该 SOC 中实现的主存储器。

[0029] 存储控制器 400，位于该 SOC 中，控制对外部存储器的访问，并且在一些实施方案

中，可以提供存储刷新机制和其他存储控制功能。在一些实施方案中，局部内存单元 300 被附接至各个处理器内核 200。每一局部内存单元可以使用直接存储器访问 (DMA) 通道通过存储控制器 400 访问外部存储器 500，例如，填充代码段 (code segment)。

[0030] 根据本发明的一些实施方案，由访问主存储器中的共享存储位置导致的性能下降通过数据流单元 (DSU)2000 得到了减轻。DSU 2000 可以被配置为，通常在处理器内核 200 需要访问来自外部存储器 500 的数据之前，用该数据预填充局部内存单元 300（每个处理器内核一个），从而使由对外部存储器的访问中的读竞争 (read contention) 导致的处理器 200 的延迟最小化。通过类似的方式，DSU 2000 向外部存储器 500 发送由处理器内核 200 写入局部内存 300 的数据，从而使由对外部存储器访问中的写竞争导致的处理器 200 的延迟最小化。

[0031] DSU 2000 包括 :DSU 仲裁器 2200，其控制对存储控制器 400 的访问 ；以及 DSU 前端单元 2100，其中每一前端单元连接至相应的局部内存 300，并连接至单个相应的处理器内核 200。

[0032] 在一些实施方案中，对共享存储位置的访问由便笺式单元 1000 操控，便笺式单元 1000 可以包括指令缓冲器 1300，每个指令缓冲器 1300 连接至相应的处理器内核 200。便笺式单元 1000 包括一个或多个便笺式随机访问存储器 (RAM)1100，以及一个或多个便笺式控制器 1200，其中每一便笺式 RAM 通过相应的便笺式控制器 1200 连接至指令缓冲器 1300。

[0033] 在一个典型的实现中，缓冲器 1300 暂时性地存储由处理器内核 200 产生的指令，直到当前指令赢得了对所请求的便笺式控制器的仲裁（根据伴随该指令的目标地址）。在图 1 的实施方案中（虽然并不是必要的），每一处理器内核 200 被连接至各自专用的指令缓冲器 1300。一旦一个指令在该指令缓冲器中被缓冲，该缓冲器就根据伴随该缓冲指令的目标地址来请求对适合的便笺式控制器 1200 的访问。当准许访问时，该指令被从该指令缓冲器发送到该便笺式控制器以用于执行。

[0034] 便笺式单元 1000 允许了通过便笺式控制器 1200 来独立执行公共共享多处理器任务，便笺式控制器 1200 被连接以通过优化的方式来执行公共共享存储器任务，卸载处理器内核 200，并通过引进有效率的内存锁机制来确保存储器一致性。这一技术是有效率的，因为所述存储器并非被物理锁定。本文公开的技术确保了通过控制器来原子地进行完整的读取 - 修改 - 写入周期。

[0035] 数据流单元 (DSU)

[0036] 图 2 是根据本发明的一个实施方案的框图，其示意性地示出了数据流单元 (DSU)2000 及其所连接的单元，包括存储控制器 400、处理器内核 200 以及局部内存 300。DSU 2000 包括 :多个 DSU 前端单元 2100，每个 DSU 前端单元 2100 服务于相应的处理器内核 200 及其关联的局部内存 300 ；以及单个 DSU 仲裁器 2200，其在由 DSU 前端单元 2100 发起的存储器访问请求之间进行仲裁。

[0037] 所述处理器内核将该 DSU 预编程，以在初始化时将数据从存储器移动至局部内存。DSU 2000 接着将来自外部存储器 500 的数据预加载至局部内存 300，从而降低了对该外部存储器的读取访问的竞争，且因此提高了多处理器系统 10 的性能。类似地，所述处理器内核将该 DSU 预编程，以将数据从所述局部内存移动至该外部存储器，从而降低了对外部存储器的写入访问的竞争。

[0038] 这一配置由于多种原因而提高了性能。例如，去往和来自外部存储器 500 的通信量可以由知道所有通信量的单个控制器 (DSU 仲裁器）优化。此外，处理器内核 200 并未被中止 (stall)，因为它们所要求的数据被提前取得，这缩短了数据访问等待时间。

[0039] 在由处理器内核 200 进行从外部存储器的地址的每一读取操作之前，是从外部存储器 500 的对应的读取操作和对局部内存 300 的对应的写入操作，这两项操作均由 DSU 前端单元 2100 进行。类似地，在由局部处理器进行的对外部存储器的地址的每一写入操作之后，是从该局部内存的对应读取操作以及随后的对该外部存储器的写入操作，这两项操作均由该 DSU 进行。

[0040] 在一些实施方案中，处理器内核 200 和外部存储器 500 之间的数据传送是由位于所述局部内存中的多个循环缓冲器进行的，下文将结合图 3 描述。每一处理器内核将数据写入相应的循环缓冲器，并且 DSU 从所述循环缓冲器读取数据并将它写入主存储器。在主存储器为外部存储器的实施方案中，这是通过外部存储控制器 400 进行的。对于读取操作，DSU 从主存储器读取数据（如果主存储器是外部存储器的话，则通过外部存储控制器），并将该数据写入所述循环缓冲器；所述处理器内核从所述循环缓冲器读取所取得的数据。

[0041] 图 3 示意性地示出了根据本发明的一个实施方案的循环缓冲器 310 的结构。在本发明的一些实施方案中，在每个局部内存中可以实现多个循环缓冲器。位于给定的局部内存 300 中的循环缓冲器由 DSU 前端单元 2100 管理，并且被连接至这一局部内存的处理器内核 200 管理。参考图 3 和在接下来的描述中，术语“读取”和“写入”是指由局部处理器 200 进行的读取操作和写入操作。

[0042] 该循环缓冲器具有：开始指针 311，其指向局部内存 300 中的缓冲器开始的位置；以及结束指针 316，其指示局部内存 300 中的最后位置。该缓冲器是循环的，并且，对于对该缓冲器的顺序访问，结束指针 316 之后的位置是开始指针 311。

[0043] 由局部处理器 200 处理的数据元由当前元指针 313 指示，该指针由特定的处理器内核指令推进（待在下文描述）。定义了工作窗口 314，其包括用于局部处理器的有效数据。如果该当前元指针达到读取指针的值，则中止该处理器，直到新的数据到来。此外，如果读取指针 315 的值等于写入指针 312 的值，则暂停从外部存储器进一步取得读取数据，直到写入存储器指针前进。

[0044] 图 4 是根据本发明的一些实施方案的框图，其示意性地示出了 DSU 前端单元 2100。该图中还描绘了附接至该 DSU 前端单元的处理器内核 200 和局部内存 300。每一 DSU 前端单元 2100 包括 DSU 缓冲器管理单元 2110、控制单元 2130、缓冲器选择多路复用器 2140 以及外部存储器访问控制 2150。DSU 前端单元 2100 被处理器内核 200 配置为执行任务列表；这样的配置可以包括对多个位于控制单元 2130 上的寄存器进行编程，其中每一寄存器可以包括指示读取、写入和数据大小的位，以及指示每一循环缓冲器 310 的缓冲器开始指针的值和缓冲器结束指针的值的位。

[0045] DSU 缓冲器管理单元 2110（其将在下文详述）管理着循环缓冲器 310。单元 2110 递增读取指针和写入指针，将缓冲器结束指针回转推进至缓冲器开始指针，并且当从该处理器内核接收 NLI(1)（待在下文描述）指令时递增当前元指针，并再次回转推进缓冲器结束指针。在当前元指针的递增导致其值等于读取指针的值的情况下，缓冲器管理单元 2110 发信号通知处理器内核 200 中止，直到接收到新的数据。

[0046] 控制单元2130在对循环缓冲器310的访问——伴随着对局部内存200的访问——之间进行仲裁。这样的仲裁可以通过，例如，旋转优先级方案(rotating priority scheme)来进行。缓冲器选择多路复用器2140从控制单元2130取得一个到所选缓冲器的指针，并将由所述缓冲器产生的地址输出到局部内存300。

[0047] 控制单元2130也控制外部存储器访问控制2150，该外部存储器访问控制2150产生读取请求和写入请求，并从DSU仲裁器2200得到读取响应。来自外部存储器访问单元2150的数据可以直接传送至局部内存300；然而，如果从外部存储器读取或向外部存储器写入的数据的大小与局部内存的数据端口的大小不同，则可以使用附加的缓冲/逻辑（未示出）。

[0048] 图5是根据本发明的一些实施方案的框图，其示意性地示出了DSU缓冲器管理单元2110。缓冲器管理单元2110被加入DSU前端单元2100中，并控制着对多个循环缓冲器的访问。在一些实施方案中，DSU缓冲器管理单元2110包括多个相同的单缓冲器管理单元(SBM)2120，其中每个SBM控制着局部内存300中的单个相应的循环缓冲器。SBM 2120可以包括：读取指针(RP)寄存器2123，其保持读取指针315的值；写入指针(WP)寄存器2122，其保持写入指针312的值；以及当前元指针(CEP)寄存器2124，其保持CEP 313的值。读取指针指向下一这样的存储位置，在该存储位置处DSU前端写入它从外部存储器读取的数据；并且，写入指针指向下一这样的存储位置，DSU前端从该位置读取它即将写入外部存储器的数据。

[0049] SBM 2120还包括：尺寸寄存器(Size Register)2121，其存储在当前数据事务中所传送的数据单元的尺寸（例如，以字节）；加法器2126，其将存储在尺寸寄存器2121中的值添加到指针寄存器2122、2123、2124中的值上，以在每一事务之后对它们进行更新；以及比较器2125，其将CEP寄存器2124的值与RP寄存器2123的值作比较，并在它们相等时断言(assert)“中止”输出，从而处理器内核200应被中止。

[0050] 读取指针寄存器2123和写入指针寄存器2122的更新由一“选择”输入来限定(qualify)，该“选择”输入由控制单元2130（见图4）激活，其中对于每一SBM 2120具有一条分立的选择线；该选择线被标记为选择-1、选择-2等。除了所述选择线，控制单元2130还对所有SBM断言公共控制线：读取和写入，以分别更新所选SBM 2120的写入指针寄存器2122和读取指针寄存器2123；以及NLI(1)，以更新DSU前端单元2100的所有SBM 2120的所有CEP寄存器2124。

[0051] 现在返回图4，控制单元2130执行在DSU前端中预编程的任务。该控制单元被处理器内核200预配置有一个存储器传送任务列表；它接着根据一个标准从缓冲器管理单元2110中选择指针，所述标准可以是，例如，旋转优先级。该控制单元接着控制缓冲器选择多路复用器(buffer-select-mux)2140的地址输入，以将所选择的地址输出至局部内存300，并向外部存储器访问控制2150发送指示词，该指示词可以包括该缓冲器的索引和读/写位。此外，当任意SBM 2120设置其“中止”输出时，该控制单元将向处理器内核200断言一个总的(aggregate)“中止”输出，以暂停其操作，直到中止被清除。

[0052] 除了在初始化时进行的DSU配置，控制单元2130还被两个特定的处理器指令控制：NLI(0)和NLI(1)。NLI(1)用于推进所述循环缓冲器中的当前元指针313。当该处理器需要新的一组参数来计算下一个值或下多个值时，在每一程序循环迭代之后设置NLI(1)。

[0053] 所更新的参数包括所有活动的 SBM 的当前元指针。通常在单个周期里对所有 SBM 进行更新，并考虑循环缓冲器回转位置。NLI 指令也验证当前元指针 (CEP) 的新位置是否已经被填充了从外部存储器取得的所需的数据。若没有，则中止该处理器内核，直到读取指针 (RP) 和当前元指针符合该需求 (即，直到 RP ＞ CEP)。

[0054] NLI(1) 被控制单元 2130 指向 DSU 缓冲器管理 2110。当程序开始时，NLI(0) 由处理器内核 200 发布。NLI(0) 指令的目的在于，验证初始 CEP 是否被有效处理 (即，所要求的数据已经被从外部存储器读取并写入局部内存)。

[0055] 在 NLI(0) 被接受之前，读取指针 (RP)315 和写入指针 (WP)312 通常不会从它们的初始位置前进。(RP 被递增是因为，它需要取得处理第一元所要求的数据。WP 未被递增是因为，CEP 仍然指向初始位置，其和 WP 指向的位置相同。处理后的数据尚不可访问，因此没有任何东西被写入外部存储器)。

[0056] 外部存储器访问控制 2150 由控制单元 2130 激活，以向 DSU 仲裁器 2200 启动读取请求和写入请求 ；它也从该 DSU 仲裁器取得读取响应，并将所读取的数据传送至局部内存。

[0057] 图 6 示意性地示出了根据本发明的一些实施方案的 DSU 仲裁器 2200 的框图。DSU 仲裁器 2200 从多个 DSU 前端单元 2100 接收读取请求和写入请求，在这些请求之间进行仲裁，计算外部存储器中相应的地址，以及将外部存储器访问请求发送至外部存储控制器 400。该 DSU 仲裁器也将从该存储控制器接收的对读取请求的响应路由回进行请求的 DSU 前端单元。在本发明的一些实施方案中，可以不需要写入响应，并且任意写入请求都被假定为被接受。在另一些实施方案中 (未在图 6 中示出)，写入响应被操控，且一个与读取响应锁存器 2210 类似的单元 (待在下文描述) 被添加至 DSU 仲裁器 2200。

[0058] 来自多个 DSU 前端单元 2100 的读取请求被锁存在读取请求锁存器 2230 中。在本发明的一些实施方案中，读取请求锁存器 2230 对于每一 DSU 前端单元包括一个索引，该索引指示了该请求对应于哪个循环缓冲器 ；并且包括一个未决位 (pending bit)，其指示了该请求是有效的且尚未被操控。来自读取请求锁存器 2230 的请求被输入读取请求仲裁单元 2250，读取请求仲裁单元 2250 使用例如旋转优先级方案在多个并发的读取请求之间进行仲裁。读取请求仲裁还清除读取请求锁存器 2230 中的选定请求的未决位。

[0059] 所选择的读取操作 (一个或多于一个，下文将解释) 被输出至读取地址计算单元 2270，其保持并更新到该外部存储器的指针 (对于每一局部内存 300 中的每一循环缓冲器，均有一个指针)，并且基于指针值以及基于与该外部存储器的组织结构有关的参数来计算该外部存储器中的地址 ；在本发明的一些实施方案中，所处理的目标是视频图像，并且这些参数可以包括图像宽度、高度和每像素的字节。来自读取地址计算 2270 的输出是对外部存储控制器 400 的读取请求。

[0060] 在本发明的一些实施方案中，对外部存储器的访问的带宽大于局部内存 300 的带宽 ；这可以由更宽的总线、更快的时钟或它们的任意结合来达成。在这样的实施方案中，可以期望并发地产生多个存储器访问。在这些实施方案中，读取请求仲裁单元 2250 将选择多个读取请求，且读取地址计算单元 2270 将同时为多个事务计算地址。

[0061] 用于写入请求的机制类似于所描述的用于读取请求的机制，并包括 ：写入请求锁存器 2220，以锁存写入请求 ；写入请求仲裁单元 2240，在未决写入请求之间进行仲裁 ；以及写入地址计算单元 2260，计算外部存储器中的地址。在本发明的一些实施方案中，写入请求

锁存器 2220、写入请求仲裁单元 2240 以及写入地址计算单元 2260 可以分别与读取请求锁存器 2230、读取请求仲裁单元 2250 以及读取地址计算单元 2270 相同。在另一些实施方案中，所述单元在性质上相似，但是由于写入不及读取频繁，所以与写入有关的单元的实现可以在较小的区域和性能方面被优化。

[0062] 最后，读取响应锁存器 2210 锁存了来自存储控制器 400 的对读取请求的响应，并将它们输出至启动这些请求的那些 DSU 前端单元 2100。

[0063] 便笺式单元

[0064] 根据本发明的一些实施方案，访问共享存储器资源的多处理器系统 10 的程序被便笺式单元 100 操控，它保证了存储器一致性，并减轻了与访问共享存储器资源相关联的延迟。便笺式单元实际上是专用处理器，其带有为有效率地执行共享存储器任务和保证存储器一致性而被优化的指令集。

[0065] 图 7 是根据本发明的一个实施方案的框图，其示意性地示出了便笺式单元 1000 及其到处理器内核 200 的接口。为处理器内核 200 定义了一组便笺式指令 (SP 指令，待在下文描述)。这些 SP 指令由处理器内核 200 发送到指令缓冲器 1300，指令缓冲器 1300 接着将它们发送至便笺式控制器 1200 以用于执行。

[0066] 如果处理器内核 200 所连接的指令缓冲器尚未将前一指令转发给便笺式控制器 1200，或者如果前一指令期待一个返回值而该返回值尚未被获得，则处理器内核 200 将避免将新的 SP 指令发送至该指令缓冲器。这一机制可以，例如，通过中止适合的处理器内核来以硬件实现。

[0067] 指令缓冲器 1300 将 SP 指令发送至两个便笺式控制器 1200 之一，该便笺式控制器 1200 在来自多个指令缓冲器 1300 的多个指令之间进行仲裁，并选择其中一个用于执行。根据本发明的一些实施方案，所述仲裁使用旋转优先级方案。

[0068] 涉及偶 (even) 便笺式存储位置的指令被输出至偶便笺式控制器 1200，而涉及奇 (odd) 便笺式存储位置的指令被输出至奇便笺式控制器 1200。因此，该指令中指定的地址的最低有效位不被转发至该便笺式控制器，而是用于选择这两个控制器之一。

[0069] 每一便笺式控制器 1200 被连接至一个 RAM 1100，该 RAM 1100 是便笺式存储器或其一部分。根据本发明的一些实施方案，便笺式存储器是根据偶地址和奇地址而交错 ；一个 RAM 1100 保存偶地址 (图 7 中的 RAM 偶)，而另一 RAM 1100(RAM 奇) 保存奇地址。连接至 RAM 偶的便笺式控制器 1200 被命名为“便笺式控制器偶”，连接至 RAM 奇的便笺式控制器 1200 被命名为“便笺式控制器奇”。在本发明的另一些实施方案中，也可以使用其他类型的交错，例如根据这两个最低有效地址位划分为四组，或者例如通过哈希函数划分。

[0070] 一些 SP 指令可以向调用处理器返回值。为此目的，每一便笺式控制器 1200 将返回值输出至所有的处理器内核 200。此外，为了使每一处理器内核确定输入数据是否是其已发布的指令的返回值，每一便笺式控制器 1200 断言 ID 总线上作为从该 RAM 返回的数据的目的地的处理器的 ID 码，该 ID 码被输出至所有的处理器内核 200。在仲裁被准许之后，该便笺式控制器中的等待时间被固定。因此，在另外一个实施方案中，指令缓冲器可以计数所述周期，并根据这一计数来抓取该数据。

[0071] 表 1 是根据本发明的一些实施方案的一个包括 9 个 SP 指令的列表。每个 SP 指令包括一个操作码——该操作码用来区别各种 SP 指令，并可以包括 4 位索引——该索引是便

笺式存储器中的地址，还可以包括，例如 16 至 18 位和一个或两个操作数。

[0072] 表 1 ：便笺式指令

[0073]

指令	操作数	解释
scp_add	**[index], [OP1]**	**便笺式添加指令，实现:** **SCP_MEM[index] =** **SCP_MEM[index] + OP1**
scp_max	**[index], [OP1]**	**便笺式最大指令，实现:** **SCP_MEM[index] =** **MAX(SCP_MEM[index], OP1)**
scp_min	**[index], [OP1]**	**便笺式最小指令，实现:** **SCP_MEM[index] =** **MIN(SCP_MEM[index], OP1)**
scp_or	**[index], [OP1]**	**便笺式按位或指令，实现:** **SCP_MEM[index] =** **SCP_MEM[index] OR OP1**
scp_and	**[index], [OP1]**	**便笺式按位与指令，实现:** **SCP_MEM[index] =** **SCP_MEM[index] AND OP1**
scp_xor	**[index], [OP1]**	**便笺式按位异或指令，实现:** **SCP_MEM[index] =** **SCP_MEM[index] XOR OP1**
scp_cax	**[index],** **[OP1], [OP2]**	**便笺式比较及交换指令，实现:** **if(SCP_MEM[index] == OP1)** **SCP_MEM[index] = OP2** **RETURN_VALUE =** **ORIGINAL_SCP_MEM[index]**
scp_ld	**[index]**	**便笺式加载指令，实现:** **RETURN_VALUE =** **SCP_MEM[index]**
scp_st	**[index], [OP1]**	**便笺式存储指令，实现:** **SCP_MEM[index] = OP1**

[0074] 图 8 是根据本发明的一些实施方案的框图，其示意性地示出了便笺式控制器 1200 的流水线级 (pipeline stages) 和结构。便笺式控制器 1200 包括 ：旋转优先级仲裁器

1210（仲裁器）、读取级单元 1220、执行级单元 1230、写入级单元 1240、比较器 1260 以及多路复用器 1250。便笺式控制器 1200 具有流水线架构，并且指令的执行在流水线级中进行。当写入级 1240 写入指令 n 的结果时，执行级 1230 进行指令 n+1 的执行部分，读取级 1220 从指令 n+2 中指定的存储器地址取得数据，并且仲裁器 1210 从指令缓冲器 1300 中取得指令 n+3。

[0075] 仲裁器 1210 在从所述指令缓冲器输入的指令之间进行仲裁，并可以采用例如旋转优先级方案。当该仲裁器选择指令资源时，它将所选择的指令转发至读取级 1220，并断言一个“就绪“输出，这向指令缓冲器指示，可以应用其下一指令（如果可以获得的话）。

[0076] 从仲裁器 1210 转发到读取级 1220 的指令可以包括五个字段，其中头四个字段是从所选择的指令拷贝而来的，包括操作码 (opcode) 字段、索引字段以及一个或两个操作数——OP1 和 OP2；上面的表 1 描绘了由每一指令使用的字段。第五个字段是一个 ID 字段，其识别所选择的指令缓冲器，并使得能够将返回值路由到原始发出该指令的处理器内核。

[0077] 在一个时钟周期的延迟后，读取级 1220 将从该仲裁器接收的指令的各字段输出至执行级 1230。此外，读取级 1220 断言读取地址总线上待从该便笺式存储器取得的操作数的地址，该读取地址总线被连接至 RAM 1100 的读取地址端口。该地址可以与从仲裁器 1210 输出的索引字段相同。从该 RAM 读取的数据经由多路复用器 1250（待在下文解释）被路由至执行级 1230；由于通过 RAM 1100 的延迟是一个时钟周期，所以来自该 RAM 的数据以及相应的指令将在相同的时钟周期到达执行级 1230。

[0078] 执行单元 1230 执行该指令，这可以包括：进行逻辑 / 算术操作（如果要求的话），输出从 RAM 1100 读取的返回值，以及激活写入级 1240。该返回值，与已经启动该指令的处理器的 I D 一起，被输出到所有的处理器内核 200。

[0079] 写入级 1240 仅在该指令具有写入部分时被激活。它从该执行级获得存储位置的索引（地址）和待写入的数据，以及写入激活信号 (writeactivation signal)。如果要求写入，则执行级 1230 断言一个写入输出，并发送该地址（索引）和该数据，在写入地址端口上断言该写入地址，并在 RAM 1100 的写入数据端口上断言该写入数据。

[0080] 如果一个指令从存储器读取数据而新数据被前一指令写入相同的位置，则存储器一致性机制被调用；由于流水线施加的一个时钟周期的延迟，所以这两次访问发生在同一时钟周期。比较器 1260 将写入地址与读取地址作比较；如果这两个地址是相同的，则多路复用器 1250 将把写入数据直接路由至执行级 1200，并且从 RAM 1100 读取的数据将被忽略。

[0081] 图 1-8 中示出的系统 10 的配置和各种系统原件的配置均是示例性配置，它们仅是为了澄清概念的目的而示出。在其他实施方案中，也可以使用其他适合的配置。在一些实施方案中，这里描述的控制器和处理器，例如存储控制器 400、处理器内核 200 以及便笺式控制器 1200，可以包括通用处理器，其被以软件编程以实现本文所描述的功能。所述软件可以例如通过网络以电子形式下载至该处理器，或者其可以，替代地或补充地，被提供和 / 或存储在非暂时性有形介质诸如磁、光或电子存储器中。

[0082] 虽然本文描述的实施方案主要针对 SOC 多处理器系统，但本文描述的方法和系统也可以被用于其他应用，诸如分布在多个集成电路中、由总线或网络或其任意结合互联的多处理器系统。

[0083] 应认识到的是，本文所描述的实施方案以实施例的形式给出，并且本发明并不限

于上文具体示出和描述的内容。相反，本发明的范围包括上述的各种特征的结合和子结合，以及本领域普通技术人员在阅读上文描述后将了解的、现有技术中未公开的变体和改型。

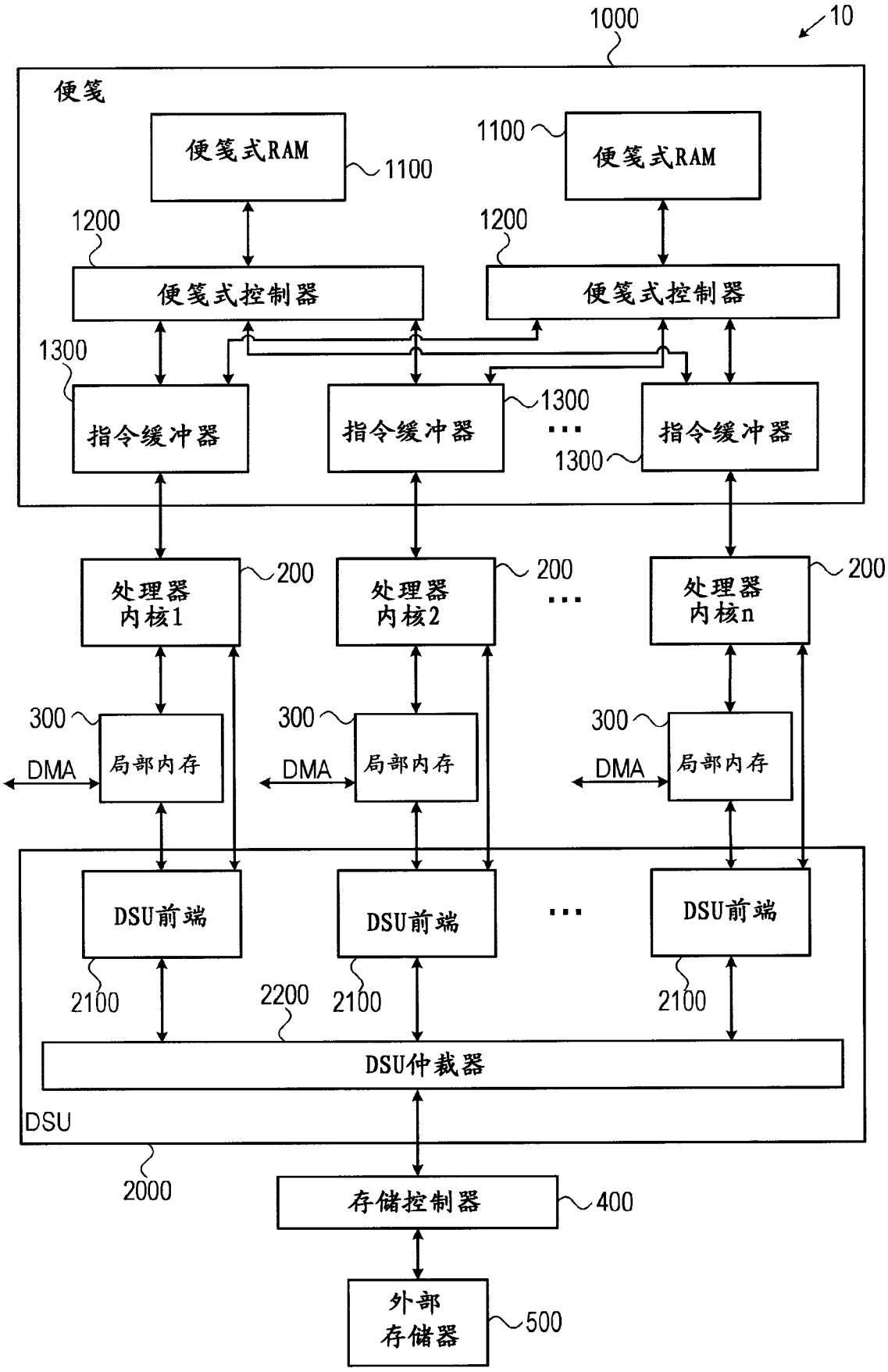

1000
10
便笺
便笺式RAM
1100
便笺式RAM
1100
1200
1200
便笺式控制器
便笺式控制器
1300
1300
1300
指令缓冲器
指令缓冲器
指令缓冲器
处理器内核1
200
处理器内核2
200
处理器内核n
200
300
300
300
DMA
局部内存
DMA
局部内存
DMA
局部内存
DSU前端
DSU前端
DSU前端
2100
2200
2100
2100
DSU仲裁器
DSU
2000
存储控制器
400
外部存储器
500

图 1

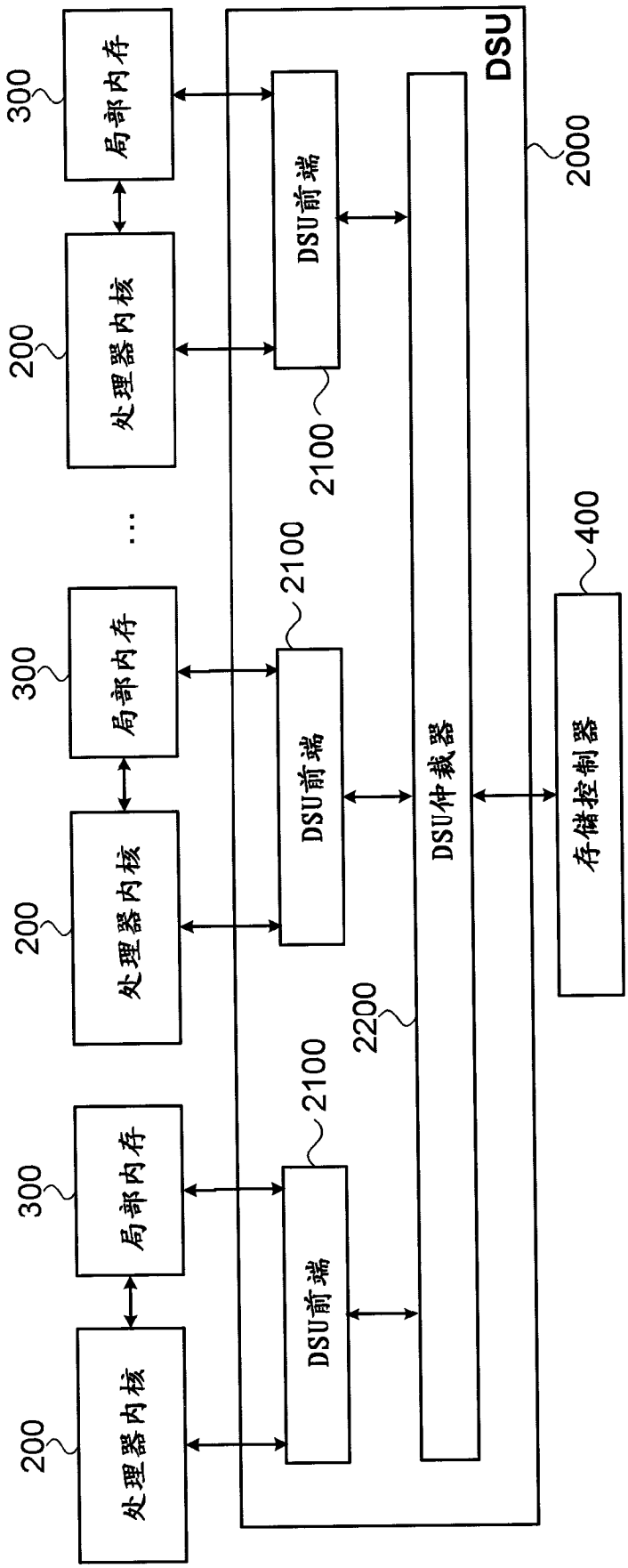

DSU
2000
处理器内核
200
局部内存
300
DSU前端
2100
DSU仲裁器
2200
存储控制器
400

图 2

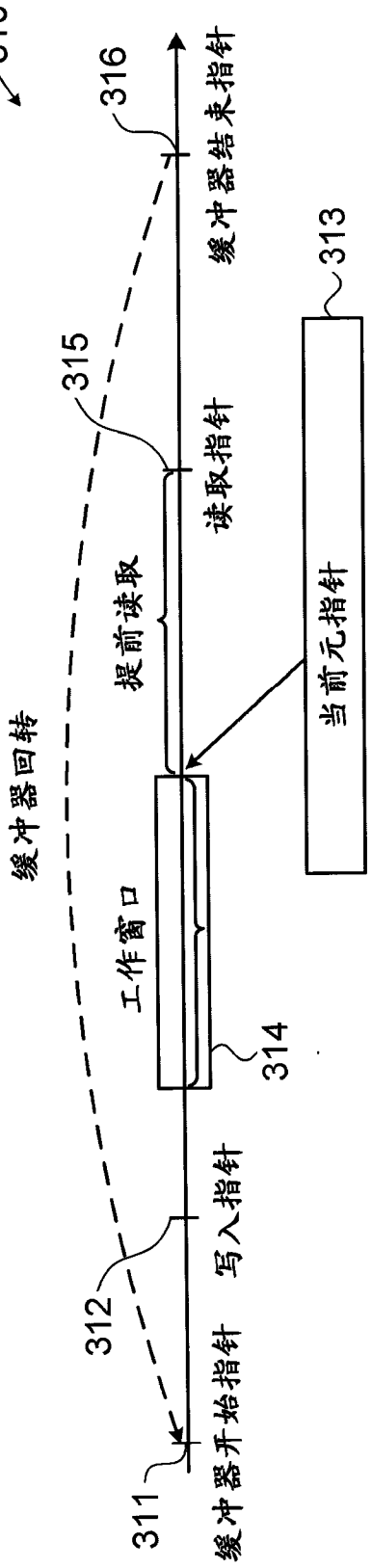

310
缓冲器回转
316
缓冲器结束指针
315
提前读取
读取指针
313
当前元指针
工作窗口
314
312
写入指针
311
缓冲器开始指针

图 3

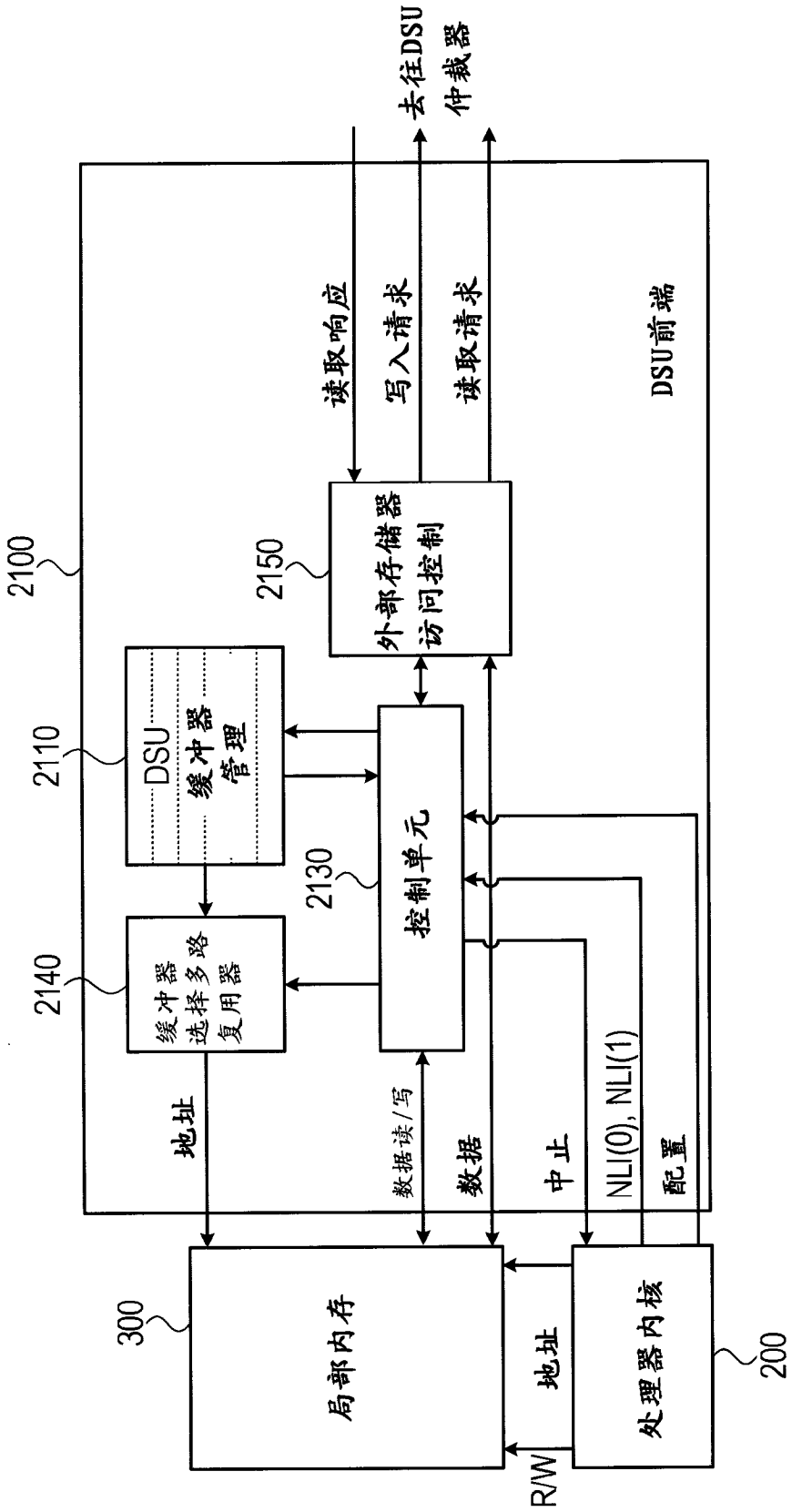

去往DSU仲裁器
读取响应
写入请求
读取请求
DSU前端
2100
2150
外部存储器访问控制
2110
DSU缓冲器管理
2130
控制单元
2140
缓冲器选择多路复用器
地址
数据读/写
数据
中止
NLI(0), NLI(1)
配置
300
局部内存
地址
R/W
处理器内核
200

图 4

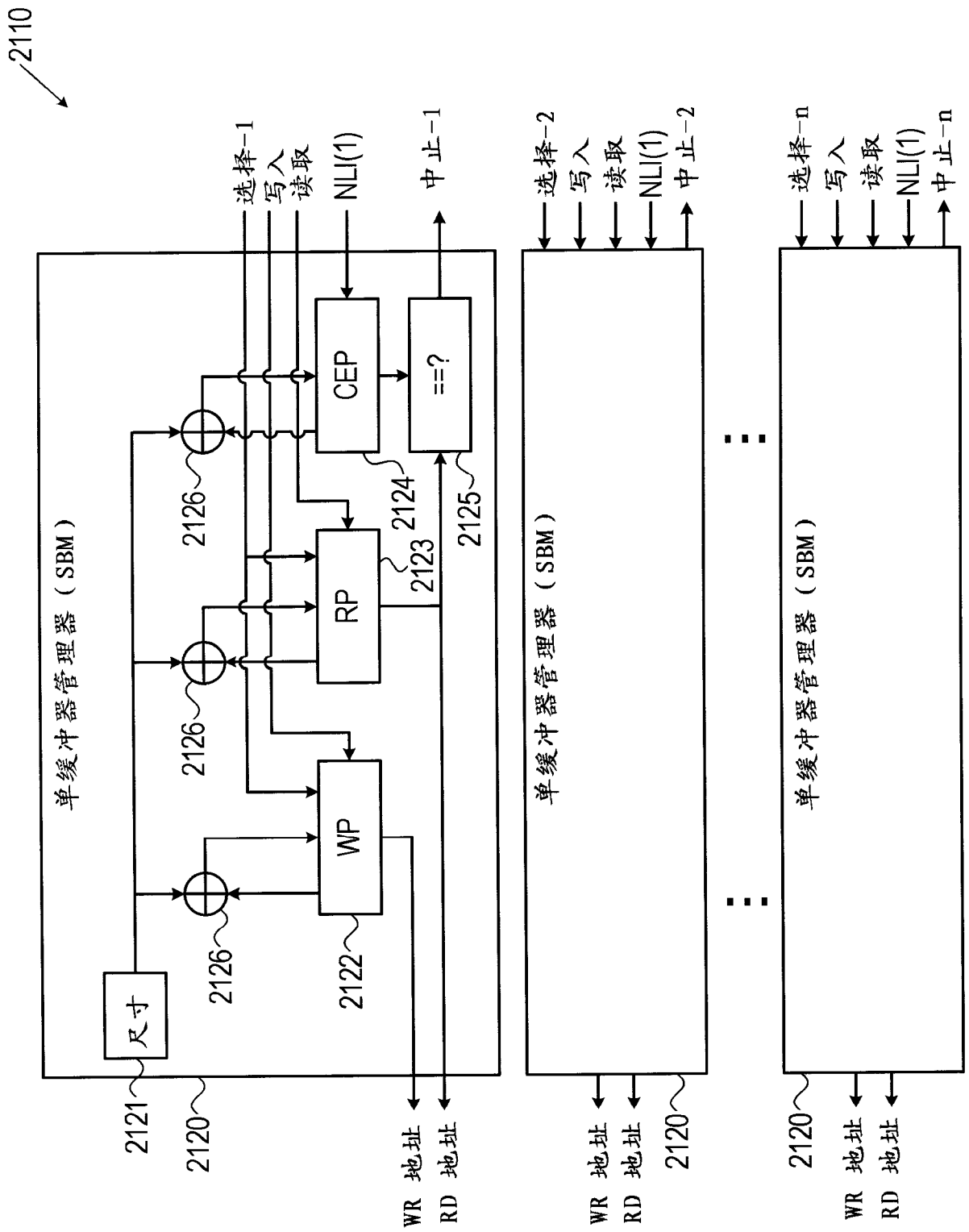

2110
单缓冲器管理器（SBM）
选择-1
写入
读取
NLI(1)
中止-1
CEP
==?
RP
WP
尺寸
2121
2120
2122
2123
2124
2125
2126
WR 地址
RD 地址
单缓冲器管理器（SBM）
选择-2
写入
读取
NLI(1)
中止-2
单缓冲器管理器（SBM）
选择-n
写入
读取
NLI(1)
中止-n

图 5

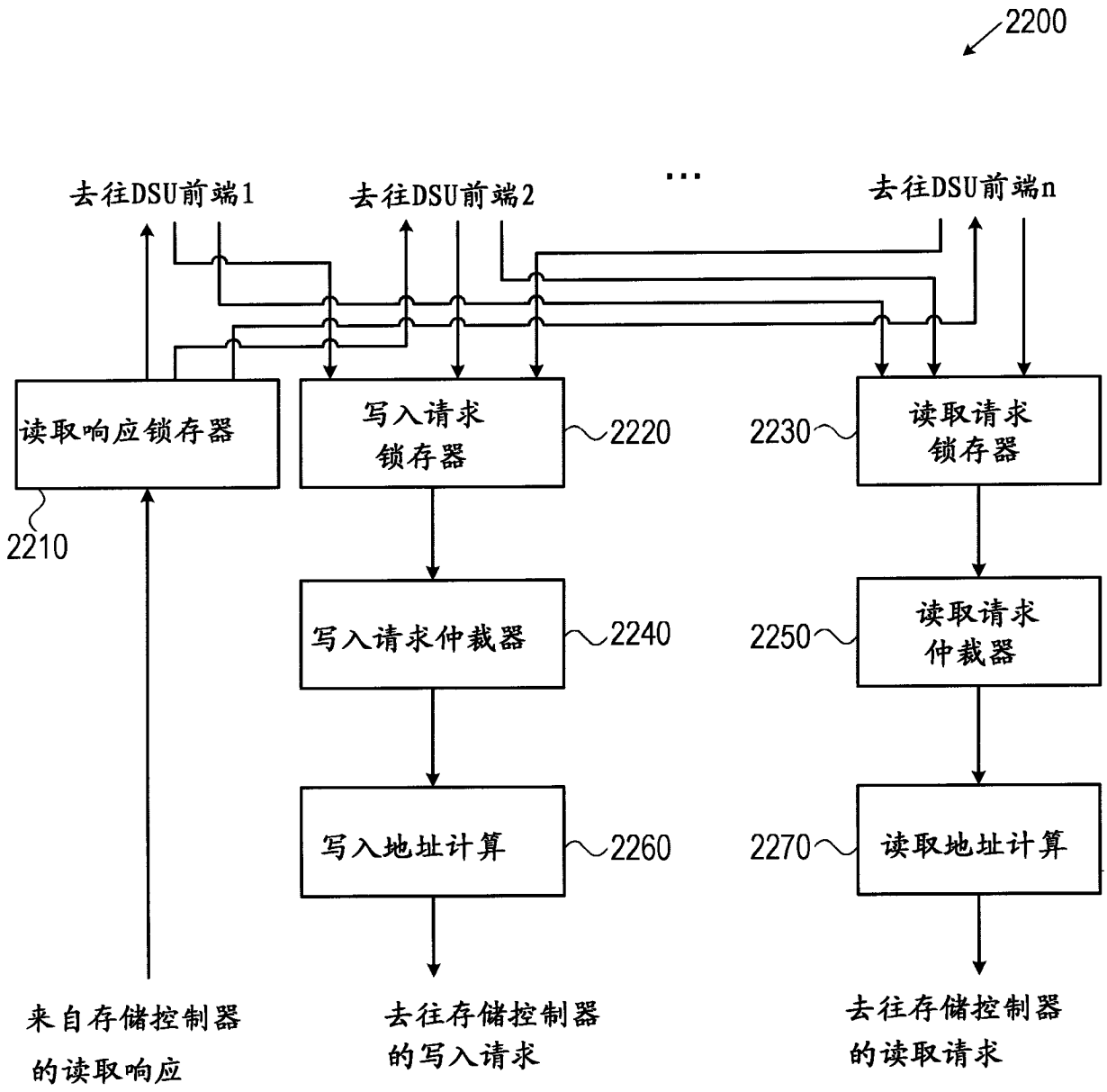

2200
去往DSU前端1
去往DSU前端2
...
去往DSU前端n
读取响应锁存器
2210
写入请求锁存器
2220
2230
读取请求锁存器
写入请求仲裁器
2240
2250
读取请求仲裁器
写入地址计算
2260
2270
读取地址计算
来自存储控制器的读取响应
去往存储控制器的写入请求
去往存储控制器的读取请求

图 6

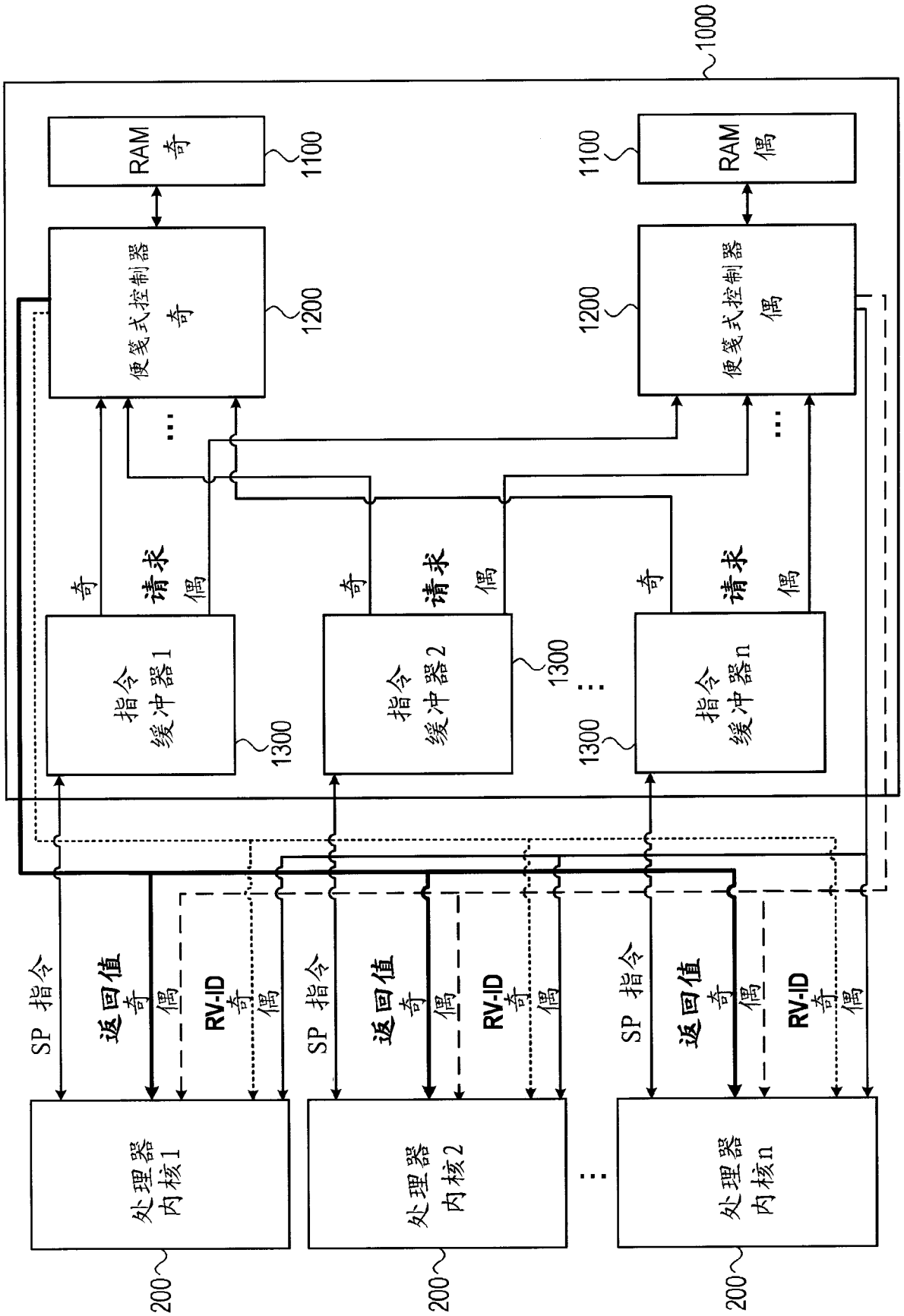

1000
RAM 奇
1100
RAM 偶
1100
便笺式控制器 奇
1200
便笺式控制器 偶
1200
奇
请求
偶
指令缓冲器1
1300
指令缓冲器2
1300
指令缓冲器n
1300
SP 指令
返回值 奇 偶
RV-ID 奇 偶
处理器内核1
处理器内核2
处理器内核n
200

图 7

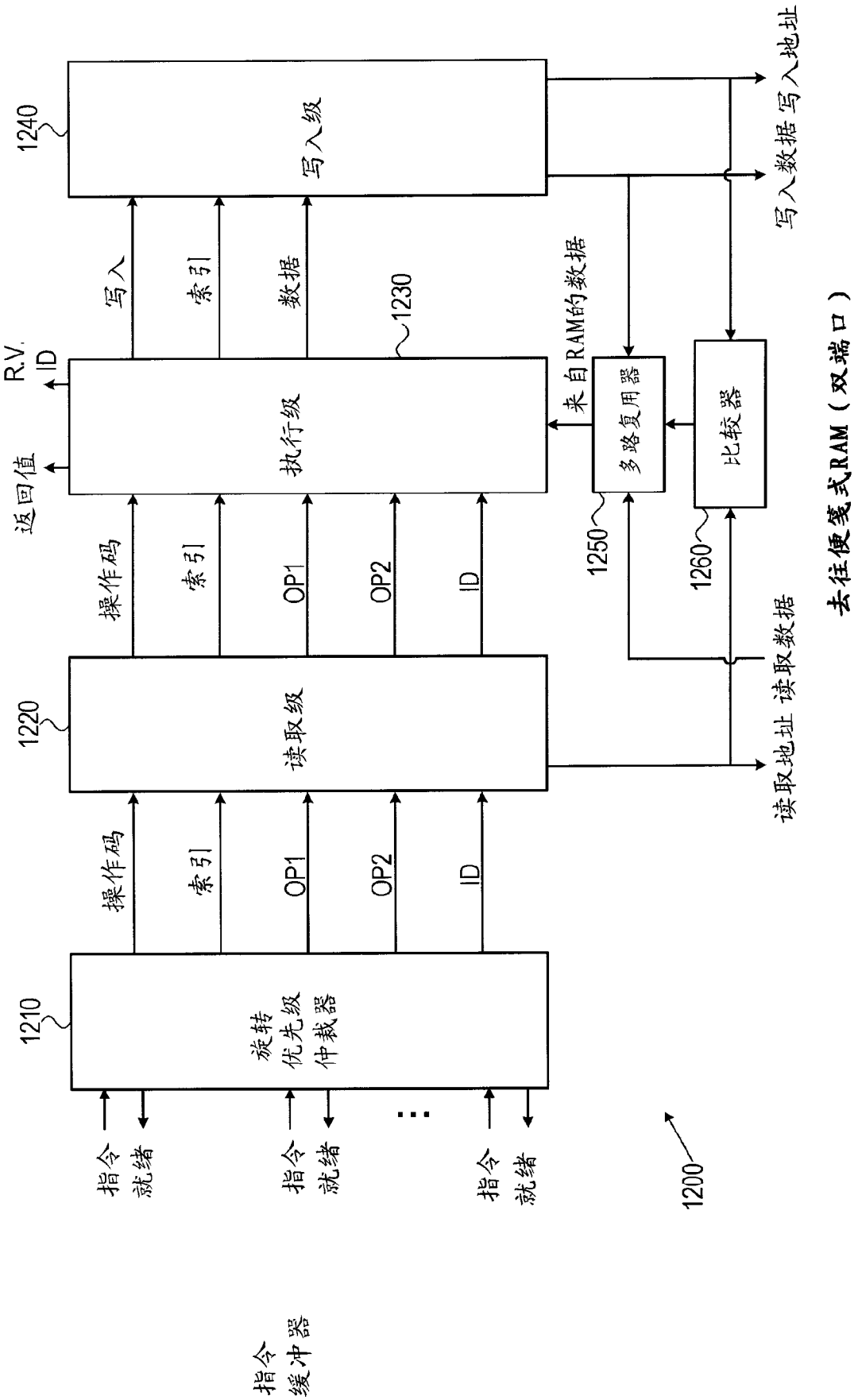

1240
写入级
写入
索引
数据
1230
R.V.
ID
返回值
执行级
来自RAM的数据
写入数据 写入地址
操作码
索引
OP1
OP2
ID
1250
多路复用器
比较器
1260
1220
读取级
读取地址 读取数据
去往便笺式RAM（双端口）
操作码
索引
OP1
OP2
ID
1210
旋转
优先级
仲裁器
指令
就绪
指令
就绪
指令
就绪
1200
指令
缓冲器

图 8