[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/16 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200810086073.1

[43] 公开日 2009 年 9 月 16 日

[11] 公开号 CN 101533386A

[22] 申请日 2008.3.14
[21] 申请号 200810086073.1
[71] 申请人 国际商业机器公司
地址 美国纽约
[72] 发明人 李 晖 王佰玲

[74] 专利代理机构 北京市中咨律师事务所
代理人 李 峥 周春燕

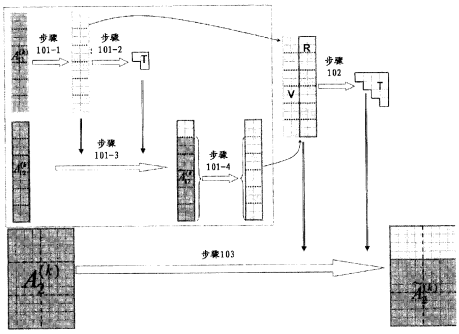
权利要求书 6 页 说明书 16 页 附图 12 页

[54] 发明名称

在多处理器系统上对矩阵进行 QR 分解的方法和装置

[57] 摘要

本发明提供了一种在多处理器系统上对矩阵进行 QR 分解的方法和装置。 其中该多处理器系统包括至少一个核心处理器以及多个加速器。 该方法包括：迭代地分解上述矩阵中的每个条带直到全部矩阵被分解；其中，在每一次迭代中包括：按照预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块；将该未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。 本发明使具有高计算能力的多处理器系统能够应用于计算任务量大的矩阵 QR 分解。



1. 一种在多处理器系统上对矩阵进行QR分解的方法，其中该多处理器系统包括至少一个核心处理器以及多个加速器（accelerator），该方法包括:

迭代地分解上述矩阵中的每个条带（panel）直到全部矩阵被分解;

其中，在每一次迭代中包括:

按照预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块;

将该未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及

利用上述多个加速器逐一在上述至少两个子条带上进行QR分解，并利用分解结果更新该至少两个子条带中还未进行QR分解的子条带的数据。

2. 根据权利要求1所述的方法，其中上述预定的块大小是64×64。

3. 根据权利要求1所述的方法，其中上述将该未处理的矩阵部分中当前处理的条带划分为至少两个子条带的步骤进一步包括:

将该未处理的矩阵部分中当前处理的条带进一步划分为左侧子条带和右侧子条带。

4. 根据权利要求3所述的方法，其中上述左侧子条带和右侧子条带的大小不等。

5. 根据权利要求3所述的方法，其中上述左侧子条带和右侧子条带的大小相等。

6. 根据权利要求3所述的方法，其中利用上述多个加速器逐一在上述至少两个子条带上进行QR分解，并利用分解结果更新该至少两个子条带中还未进行QR分解的子条带的数据的步骤进一步包括:

在上述左侧子条带上进行QR分解运算;

根据上述QR分解运算的结果，计算上述左侧子条带的三角因子;

利用上述左侧子条带及其三角因子更新上述右侧子条带的数据；以及

在更新后的上述右侧子条带上进行 QR 分解运算。

7. 根据权利要求 1 或 6 所述的方法，在每一次迭代中还包括:

在上述至少两个子条带均进行了 QR 分解运算之后，计算作为该至少两个子条带的整体的、上述当前处理的条带的三角因子；以及

利用上述当前处理的条带及其三角因子更新上述未处理的矩阵部分中还未进行迭代的矩阵部分的数据。

8. 根据权利要求 1 所述的方法，其中利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据的步骤还包括:

以行列周期性块划分的方式，将该 QR 分解运算所需的矩阵数据从该多处理器系统的主存储器分发给上述多个加速器。

9. 一种在多处理器系统上对矩阵进行 QR 分解的方法，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该方法包括:

迭代地分解上述矩阵中的每个条带直到全部矩阵被分解;

其中，在每一次迭代中包括:

判断上述矩阵中未处理的矩阵部分的维数是否小于第一阈值，若是，则:

按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块；以及

不启动上述多个加速器，而利用上述核心处理器对该未处理的矩阵部分中当前处理的条带进行 QR 分解，其中该当前处理的条带包括多个块;

否则，判断上述未处理的矩阵部分的维数是否大于上述第一阈值而小于第二阈值，若是，则:

按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块;

将对上述未处理的矩阵部分中当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速

器，其中该当前处理的条带包括多个块；以及

协调上述多个加速器的每一个从其本地或其他的加速器获取所分发的数据，以进行上述当前处理的条带的 QR 分解；

否则：

按照第二预定的块大小，将上述未处理的矩阵部分划分为多个块；

将上述未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及

利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

10. 根据权利要求 9 所述的方法，其中上述第一阈值是根据上述多个加速器之间的通信带宽的大小而确定的值，上述第二阈值是根据上述多个加速器的本地存储容量的总计而确定的值。

11. 根据权利要求 9 或 10 所述的方法，其中上述第一阈值是 256，上述第二阈值是 2K。

12. 根据权利要求 9 所述的方法，其中上述第一预定的块大小是 32 × 32，上述第二预定的块大小是 64 × 64。

13. 根据权利要求 9 所述的方法，其中将对上述未处理的矩阵部分中当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器的步骤进一步包括：

以行列周期性块划分的方式，将该矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器。

14. 根据权利要求 9 所述的方法，其中协调上述多个加速器的每一个从其本地或其他的加速器获取所分发的数据，以进行上述当前处理的条带的 QR 分解的步骤进一步包括：

为上述多个加速器的每一个，判断其进行计算所需的数据是否存在于该加速器本地；

为上述多个加速器中进行计算所需的数据不存在于该加速器本地的加速器，搜索其他加速器以获取该所需的数据；以及

协调上述多个加速器利用从其本地或其他的加速器获取的数据进行上述当前处理的条带的 QR 分解。

15. 一种在多处理器系统中对矩阵进行 QR 分解的装置，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该装置迭代地分解上述矩阵中的每个条带直到全部矩阵被分解，其包括：

块划分单元，用于在每一次迭代中，按照预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块；

条带划分单元，用于在每一次迭代中，将上述未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及

子条带处理单元，用于在每一次迭代中，利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

16. 根据权利要求 15 所述的装置，其中上述条带划分单元将上述当前处理的条带进一步划分为左侧子条带和右侧子条带；并且

上述子条带处理单元进一步包括：

子条带 QR 分解单元，用于利用上述多个加速器依次在上述左侧子条带和上述右侧子条带上进行 QR 分解运算；

子条带三角因子计算单元，用于在上述左侧子条带的 QR 分解运算结束后，根据该 QR 分解运算的结果，计算该左侧子条带的三角因子；以及

子条带更新单元，用于利用该左侧子条带及其三角因子更新上述右侧子条带的数据；

其中上述子条带 QR 分解单元在更新后的上述右侧子条带上进行 QR 分解运算。

17. 根据权利要求 15 或 16 所述的装置，还包括：

三角因子计算单元，用于在每一次迭代中，在上述至少两个子条带均

进行了 QR 分解运算之后，协调上述多个加速器计算作为该至少两个子条带的整体的、上述当前处理的条带的三角因子；以及

矩阵更新单元，用于在每一次迭代中，利用上述当前处理的条带及其三角因子更新上述未处理的矩阵部分中还未进行迭代的矩阵部分的数据。

18. 一种在多处理器系统中对矩阵进行 QR 分解的装置，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该装置迭代地分解上述矩阵中的每个条带直到全部矩阵被分解，其包括：

常规 QR 分解单元，用于按照第一预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块，并且利用上述核心处理器对其中当前处理的条带进行 QR 分解，其中该当前处理的条带包括多个块；

第一方案模块，用于按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块，并将对其中当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器，并使上述多个加速器的每一个从其本地或其他的加速器获取数据进行上述当前处理的条带的 QR 分解，其中该当前处理的条带包括多个块；

第二方案模块，用于按照第二预定的块大小，将上述未处理的矩阵部分划分为多个块，并且将其中当前处理的条带划分为至少两个子条带，利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据，其中该当前处理的条带包括多个块；

选择单元，用于在每一次迭代中判断上述矩阵中未处理的矩阵部分的维数是否小于第一阈值，如果是，则针对该未处理的矩阵部分启动上述常规 QR 分解单元；否则，判断该未处理的矩阵部分的维数是否大于该第一阈值而小于第二阈值，如果是，则针对该未处理的矩阵部分启动上述第一方案模块；否则，针对该未处理的矩阵部分启动上述第二方案模块。

19. 根据权利要求 18 所述的装置，其中上述第一方案模块进一步包括：

数据分发单元，用于将对上述当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器；

判断单元，用于为上述多个加速器的每一个，判断其进行计算所需的数据是否存在于该加速器本地；

数据获取单元，用于为上述多个加速器中进行计算所需的数据不存在于该加速器本地的加速器，搜索其他加速器以获取该所需的数据；以及

QR 分解单元，用于协调上述多个加速器利用从其本地或其他的加速器获取的数据进行上述当前处理的条带的 QR 分解、三角因子的计算，并根据计算结果更新其余的矩阵部分。

在多处理器系统上对矩阵进行QR分解的方法和装置

技术领域

本发明涉及数据处理领域，具体地，涉及在多处理器系统上对矩阵进行QR分解的方法和装置。

背景技术

LAPACK（Linear Algebra PACKage，线性代数包）是Oak Ridge国家实验室、加州大学Davis分校和Illinois大学等联合开发的非常有效、强大且广泛使用的线性代数函数库，用于在不同高性能计算环境上高效求解数值线性代数问题，其已经有效地为HPC（High Performance Computing，高性能计算）和计算科学组织工作了20多年的时间。关于LAPACK的详细内容，可参见http://netlib.amss.ac.cn/lapack/index.html。

作为专业的线性代数库，LAPACK提供了各种线性代数的子程序，其中包括实现矩阵的QR分解的例程。

矩阵的QR分解的含义是：对于给定的M×N矩阵A，找出分解

$$A=Q*R,$$

其中，Q是M×M正交矩阵，R是M×N上三角矩阵。

现有的LAPACK 中的QR分解例程是采用条带（panel）QR分解方案来实现的，其是一种分块的分解方案。

图1是现有的条带QR分解方案的图示说明，其中，图1(a)和(b)分别是现有的条带QR分解方案中第k次迭代运算的总体和分步图示说明，图1(c)是现有的条带QR分解方案的算法描述。图2是现有的条带QR分解方案的流程图。

总体来说，如图1(a)所示，现有的条带QR分解方案的思路是：对于

给定的 M × N 矩阵 A，通过迭代地每次在矩阵的一个条带上进行分解运算来最终将该矩阵 A 分解为 M × M 正交矩阵 Q 和 M × N 上三角矩阵 R 的积。本发明为了简便起见，在图中的矩阵 A 都采用方阵作为图例，实际上，图中矩阵 A 可以不是方阵，而是任意 M × N 矩阵，其中，M 和 N 为不相等的正整数。以其中一次迭代为例，如图 1(a)的左侧所示，浅灰色的矩阵部分 V、R 是通过第 1~k-1 次迭代运算已经分解了的矩阵部分，而深灰色的矩阵部分 $A_1^{(k)}$和 $A_2^{(k)}$结合起来的矩阵部分则是未分解的部分，也是第 k（k=1,2,3...）次迭代运算的对象。进而，在第 k 次迭代运算中，将该深灰色矩阵部分划分为两个条带 $A_1^{(k)}$和 $A_2^{(k)}$，其中 $A_1^{(k)}$作为当前工作条带；然后，在当前工作条带 $A_1^{(k)}$上进行 QR 分解计算，并利用其分解计算的结果更新 $A_2^{(k)}$，从而得到图 1(a)右侧的矩阵。其中，在该图 1(a)右侧的矩阵中，深灰色矩阵部分 $\tilde{A}_2^{(k)}$成为第 k+1 次迭代运算的分解对象。

具体地，如图 1(b)、(c)和图 2 所示，在现有的条带 QR 分解方案中，对于给定的 M × N 矩阵 A，首先将其划分为 m × n 块，其中每一个块的大小为 $N_b \times N_b$，例如 32 × 32，然后根据

$$A^{(k)} = \begin{pmatrix} A_1^{(k)} & A_2^{(k)} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = Q \cdot \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

在第 k（k=1,2,3...）次迭代运算中执行以下步骤 1~3:

在步骤 1，从此次迭代运算的对象矩阵 $A^{(k)}$划分出由 $m \times n_b$ 块构成的条带 $A_1^{(k)}$，作为当前工作条带，并在该当前工作条带 $A_1^{(k)}$上进行 QR 分解计算，以将其分解为 V 部分和 R 部分；

在步骤 2，根据步骤 1 的计算结果，计算当前工作条带 $A_1^{(k)}$的三角因子 T；

在步骤 3，将当前工作条带 $A_1^{(k)}$以及 $A_1^{(k)}$的三角因子 T 应用于 $A^{(k)}$的剩余矩阵部分 $A_2^{(k)}$，以更新其数据。LAPACK 只输出 V 和 R 矩阵，用户可以通过 V 矩阵，计算获得 Q 矩阵，从而完成 QR 分解。

图 3 示出了利用上述现有的条带 QR 分解方案对一个被划分为 3 × 3 块的矩阵进行 QR 分解的过程（只有一次迭代的情况）。其中，如图 3(a)

所示，在第 1 步，在该矩阵左侧 3×1 块的当前工作条带上进行 QR 分解；如图 3(b)所示，在第 2 步，计算该当前工作条带的三角因子 T_k；如图 3(c)所示，在第 3 步，利用当前工作条带及其三角因子 T_k 对剩余的 3×2 块的矩阵部分进行更新。

根据上述现有的条带 QR 分解方案而设计的 QR 分解例程，会存在大量的矩阵乘操作，对于这样的例程而言，性能是非常关键的。

Cell 宽频引擎（Cell Broadband Engine，CBE）是一种单芯片多处理器系统。如图 4 所示，CBE 系统具有在一个共享的、相干的存储器上进行操作的 9 个处理器，其中包括一个主处理器（Power Processing Unit，PPU）和 8 个协处理器（Synergistic Processing unit，SPU）。在这样的系统结构下，CBE 能够提供杰出的计算能力。具体来说，Cell 处理器在时钟频率 3.2GHz 的情况下能够达到 204G 浮点运算数/秒。具有这样高的计算能力，对于高计算任务量的矩阵 QR 分解来说，CBE 显然是一个理想的执行平台。

但是，上述现有的条带 QR 分解方案是为单处理器系统而设计的，如果将其直接应用于 CBE 这样的多处理器系统，则会存在存储器带宽限制的问题。因为，在 CBE 中，每个 SPU 的本地存储器容量是 256K，这样，在超过 256K 的大数据量的情况下，就需要利用 DMA 方式在主存储器与 SPU 的本地存储器之间重复地进行读入/读出操作。例如，在将矩阵划分为每个块的大小为 32×32 的多个块的情况下，如果在 CBE 的 8 个 SPU 上实现上述现有的条带 QR 分解方案，则最大存储器需求将是 20.6GB/秒。但是，CBE 中的 QS20 和 QS21 刀片仅能够维持大致 20.5GB/秒的存储器带宽。因此，存储器带宽成为上述现有的条带 QR 分解方案应用于 CBE 那样的多处理器系统以改进 QR 分解性能的一个瓶颈。

因此，需要设计出一种适合于 CBE 这样的多处理器系统的 QR 分解方案。

发明内容

鉴于上述问题，本发明提供了一种在多处理器系统上对矩阵进行 QR

分解的方法和装置，以便利用 CBE 这样的多处理器系统来执行计算任务量较大的矩阵 QR 分解运算，从而充分发挥这样的多处理器系统所具有的高计算能力的优点。

根据本发明的一个方面，提供了一种在多处理器系统上对矩阵进行 QR 分解的方法，其中该多处理器系统包括至少一个核心处理器以及多个加速器（accelerator），该方法包括：迭代地分解上述矩阵中的每个条带（panel）直到全部矩阵被分解；其中，在每一次迭代中包括：按照预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块；将该未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

根据本发明的另一个方面，提供了一种在多处理器系统上对矩阵进行 QR 分解的方法，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该方法包括：迭代地分解上述矩阵中的每个条带直到全部矩阵被分解；其中，在每一次迭代中包括：判断上述矩阵中未处理的矩阵部分的维数是否小于第一阈值，若是，则：按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块；以及不启动上述多个加速器，而利用上述核心处理器对该未处理的矩阵部分中当前处理的条带进行 QR 分解，其中该当前处理的条带包括多个块；否则，判断上述未处理的矩阵部分的维数是否大于上述第一阈值而小于第二阈值，若是，则：按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块；将对其中当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器，其中该当前处理的条带包括多个块；以及协调上述多个加速器的每一个从其本地或其他的加速器获取所分发的数据，以进行上述当前处理的条带的 QR 分解；否则：按照第二预定的块大小，将上述未处理的矩阵部分划分为多个块；将其中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及利用上述多个加速器逐一在上述

至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

根据本发明的另一个方面，提供了一种在多处理器系统中对矩阵进行 QR 分解的装置，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该装置迭代地分解上述矩阵中的每个条带直到全部矩阵被分解，其包括：块划分单元，用于在每一次迭代中，按照预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块；条带划分单元，用于在每一次迭代中，将上述未处理的矩阵部分中当前处理的条带划分为至少两个子条带，其中该当前处理的条带包括多个块；以及子条带处理单元，用于在每一次迭代中，利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

根据本发明的另一个方面，提供了一种在多处理器系统中对矩阵进行 QR 分解的装置，其中该多处理器系统包括至少一个核心处理器以及多个加速器，该装置迭代地分解上述矩阵中的每个条带直到全部矩阵被分解，其包括：常规 QR 分解单元，用于按照第一预定的块大小，将上述矩阵中未处理的矩阵部分划分为多个块，并且利用上述核心处理器对其中当前处理的条带进行 QR 分解，其中该当前处理的条带包括多个块；第一方案模块，用于按照第一预定的块大小，将上述未处理的矩阵部分划分为多个块，并将对其中当前处理的条带进行 QR 分解所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器，以使上述多个加速器的每一个从其本地或其他的加速器获取数据进行上述当前处理的条带的 QR 分解，其中该当前处理的条带包括多个块；第二方案模块，用于按照第二预定的块大小，将上述未处理的矩阵部分划分为多个块，并且将其中当前处理的条带划分为至少两个子条带，利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据，其中该当前处理的条带包括多个块；选择单元，用于在每一次迭代中判断上述矩阵中未处理的矩阵部分的维数是否

小于第一阈值，如果是，则针对该未处理的矩阵部分启动上述常规 QR 分解单元；否则，判断该未处理的矩阵部分的维数是否大于该第一阈值而小于第二阈值，如果是，则针对该未处理的矩阵部分启动上述第一方案模块；否则，针对该未处理的矩阵部分启动上述第二方案模块。

附图说明

相信通过以下结合附图对本发明具体实施方式的说明，能够使人们更好地了解本发明上述的特点、优点和目的。

图 1 是现有的条带 QR 分解方案的图示说明；

图 2 是现有的条带 QR 分解方案的流程图；

图 3 示出了利用现有的条带 QR 分解方案对一个 3×3 块的矩阵进行 QR 分解的过程；

图 4 是 CBE 的系统框图；

图 5 是根据本发明实施例的在多处理器系统上对矩阵进行 QR 分解的方法的流程图；

图 6 是根据本发明实施例的在多处理器系统上对矩阵进行 QR 分解的方法的图示说明；

图 7 是图 5 中的第一方案的流程图；

图 8 示出了对矩阵进行划分的几种方式；

图 9 是各个 SPU 的本地存储器被分为两部分的 CBE 的系统框图；

图 10 是图 5 中的第二方案的流程图；

图 11 是图 5 中的第二方案的图示说明；

图 12 示出了利用本发明的第二方案对一个 3×3 块的矩阵进行 QR 分解的过程；

图 13 是根据本发明实施例的在多处理器系统中对矩阵进行 QR 分解的装置的方框图；以及

图 14 是图 13 中的第二方案模块的方框图。

具体实施方式

下面就结合附图对本发明的各个优选实施例进行详细说明。

图 5 是根据本发明实施例的在多处理器系统上对矩阵进行 QR 分解的方法的流程图。其中，该多处理器系统具有至少一个核心处理器以及多个加速器（accelerator）。具体地，该多处理器系统例如可以是前述具有一个 PPU（核心处理器）和 8 个 SPU（加速器）的 CBE。

本实施例的在多处理器系统上对矩阵进行 QR 分解的方法，与前述现有的条带 QR 分解方案同样，对于给定的 M × N 矩阵 A，通过迭代地每次在矩阵的一个条带上进行分解运算来最终将该矩阵 A 分解为 M × N 矩阵 V 和 M × N 上三角矩阵 R 的积，再通过计算获得 M × M 的矩阵 Q，从而完成 QR 分解。其中，如图 6 的左侧所示，浅灰色的矩阵部分 V、R 是通过第 1~k-1 次迭代运算已经分解了的矩阵部分，而深灰色的矩阵部分 $A_1^{(k)}$和 $A_2^{(k)}$结合起来的矩阵部分则是未分解的部分，也是第 k（k=1,2,3...）次迭代运算的对象。并且，图 6 右侧的矩阵是第 k 次迭代运算之后的矩阵，其中，深灰色的矩阵部分 $\tilde{A}_2^{(k)}$成为第 k+1 次迭代运算的分解对象。

具体地，本实施例的在多处理器系统上对矩阵进行 QR 分解的方法，在第 k（k=1,2,3...）次迭代运算中执行以下步骤 505~525。

如图 5 所示，在步骤 505，对于上述 M × N 矩阵 A 中未处理的矩阵部分，即本次迭代运算的对象，判断其维数是否小于第一阈值。如果是，则转到步骤 510，否则前进到步骤 515。

其中，该第一阈值是根据上述多处理器系统中的多个加速器之间的通信带宽的大小而确定的。在本实施例中，其例如可以是 256。

在步骤 510，上述未处理的矩阵部分的维数小于第一阈值，表明其成为较小型矩阵，所以仅使用该多处理器系统中的核心处理器（在 CBE 的情况下为 PPU）对其进行 QR 分解。其中，该 QR 分解可以按照前述现有的条带 QR 分解方案进行，即首先将该未处理的矩阵部分划分为多个块，其中每个块的大小可以是 32 × 32；然后从中划分出由多个块构成的当前工作条带，并在其上进行 QR 分解；然后利用该当前工作条带的分解结果更新

其余的矩阵数据。

此外，在本实施例中，之所以要对小于 256 维的较小型矩阵仅启用核心处理器而不启用加速器，是基于这样的考虑：完成小于 256 维这样的较小型矩阵的 QR 计算所需的时间很短，而启动多个诸如 SPU 的加速器也需要一定的时间，权衡之下若启动加速器则并不能够在这样较小型矩阵的情况下带来计算性能的显著提高。

此外，需要说明的是，在本实施例中，虽然以 256 维作为衡量未处理的矩阵部分是否成为较小型矩阵的标准，但是，本领域技术人员应该能够理解，这仅是示例性而并非限制性的，根据本说明书的教导，在具体实现中可以依情况采用任何其他适合的值作为衡量较小型矩阵的标准。

接着，在步骤 515，在上述未处理的矩阵部分的维数大于第一阈值的情况下，判断其维数是否小于第二阈值。如果是，则转到步骤 520，否则前进到步骤 525。

其中，该第二阈值是根据上述多个加速器的本地存储器的容量总计而确定的值。更具体地，该第二阈值是基于在进行 QR 分解时能够将一次迭代运算所需的矩阵数据全部分发到该多个加速器的本地存储器中、而无需使该多个加速器在一次迭代运算的过程中从主存储器读取数据的考虑而设定的。例如，在具有 8 个 SPU 的 CBE 的情况下，由于每个 SPU 的本地存储器容量是 256K 字节，这 8 个 SPU 的本地存储器的容量总计将是 256K*8=2048K 字节，所以可以将该第二阈值设定为 2K，使一次迭代运算所需的数据能够完全分布在这 8 个 SPU 的本地存储器中。

当然，本领域技术人员应该能够理解，2K 仅是示例性而并非限制性的，根据本说明书的教导，在具体实现中可以依情况采用任何其他适合的值。

在步骤 520，采用图 7 所示的第一方案，对该维数大于第一阈值而小于第二阈值的未处理矩阵部分进行 QR 分解。

在步骤 525，上述未处理的矩阵部分的维数大于第二阈值，表明其是较大型矩阵，因而采用图 10 所示的第二方案，对其进行 QR 分解。

图 7 是根据本发明实施例的在多处理器系统上对矩阵进行 QR 分解的

第一方案的流程图。

本实施例的该第一方案用于在诸如 CBE 的多处理器系统上对维数大于第一阈值、例如 256 而小于第二阈值、例如 2K 的矩阵进行当前工作条带的 QR 分解。

具体地，本实施例的该第一方案，在一次迭代运算中，如图 7 所示，首先在步骤 705，将上述未处理的矩阵部分划分为多个块，其中每个块的大小可以是 32 × 32。

然后，在步骤 710，与前述现有的条带 QR 分解方案同样，从上述未处理的矩阵部分中划分出由多个块构成的当前工作条带，以便在其上进行 QR 分解。但是，所不同的是，本实施例的该第一方案是利用多个加速器来共同完成 QR 分解这一过程的，因此，在进行分解之前，首先要进行分解所需的数据的分发等步骤 715~725。

在步骤 715，将上述当前工作条带的 QR 分解运算所需的矩阵数据从该多处理器系统的主存储器全部分发到上述多个加速器的本地存储器中。

由于如上所述，第二阈值是基于在进行 QR 分解时能够将一次迭代运算所需的矩阵数据全部分发到该多个加速器的本地存储器中、而无需使该多个加速器在一次迭代运算的过程中从主存储器读取数据的考虑而设定的，所以在该第二阈值的确保下，能够将上述当前工作条带的 QR 分解运算所需的数据全部分布在该多个加速器的本地存储器中。

此外，为了实现矩阵数据的分发，图 8(a)~(e)示出了对需要分发的矩阵部分进行划分（图 8 中所示出的是划分为 4 部分以分发给 4 个加速器的情况）的几种方式。其中，具有相同标记的矩阵部分将分发给同一加速器。具体地，图 8(a)所示出的是列块划分方式，即按照加速器的数量将所需要分发的矩阵部分划分为均等列块；图 8(b)所示出的是按列周期性划分方式；图 8(c)所示出的是按列周期性块划分方式；图 8(d)所示出的是行列周期性块划分方式；图 8(e)所示出的是块倾斜划分（block skewed layout）方式。

在本实施例中，优选使用图 8(d)所示出的行列周期性块划分方式对当前工作条带的 QR 分解计算所需要的矩阵数据进行划分以分发给上述多个

加速器，由多个加速器同时对当前条带进行 QR 分解。当然，并不限于此，在具体实现中，也可以依情况采用图 8(a)、(b)、(c)或(e)所示出的方式。

接着，在步骤 720，判断各个加速器进行计算所需的数据是否存在于其本地存储器内。如果不存在，则前进到步骤 725，否则转到步骤 730。

在该第一方案中，由于利用多个加速器来共同执行当前工作条带的 QR 分解，所以每个加速器都要分担其中的一部分数据的计算，所以在各个加速器进行自身的计算之前，首先需要确保这各个加速器所负责计算的数据部分存在于其本地存储器内。

在步骤 725，对于计算数据不存在于其本地存储器内的加速器，利用 DMA 方式搜索其他加速器的本地存储器以获取所需的计算数据。

在本发明的一个实施例中，如图 9 所示，可以将诸如 SPU 的每个加速器的本地存储器划分为 A、B 两个部分，以分别存储从该多处理器系统的主存储器分发来的矩阵数据和通过 DMA 方式从其他加速器的本地存储器读取来的矩阵数据。

在步骤 730，协调上述多个加速器利用从本地存储器或其他加速器的本地存储器获取的数据进行上述当前工作条带的 QR 分解运算。

在步骤 735，根据步骤 730 的计算结果，协调上述多个加速器计算当前工作条带的三角因子。

在步骤 740，利用当前工作条带以及当前工作条带的三角因子更新上述未处理的矩阵部分中、除了该当前工作条带之外的其余矩阵部分。

其中，在步骤 730 和 735 的计算过程中，在这各个加速器相互之间要进行计算结果的实时互通，以确保计算的统一性。

以上就是对本实施例的在多处理器系统上对矩阵进行 QR 分解的第一方案的详细描述。在该第一方案中，在矩阵维数小于第二阈值的情况下，将当前工作条带的 QR 分解计算所需的矩阵数据全部分发到各个加速器的本地存储器中，并且当所需数据不在本加速器的本地存储器中时，不是从系统主存中通过 DMA 读取数据，而是通过 DMA 从其他加速器的本地存储器读取数据。从而，由于诸如 SPU 的加速器之间互连的带宽为 204.8GB/

秒，其远大于 SPU 到主存储器的带宽 25.6GB/秒，所以能够大幅度降低 QR 分解中的 DMA 开销，能够避免在 QR 分解过程中存储器带宽需求大于系统所能够提供的存储器带宽的问题。

图 10 是根据本发明实施例的用于在多处理器系统上对矩阵进行 QR 分解的第二方案的流程图，图 11 是该第二方案的图示说明。

本实施例的该第二方案用于在诸如 CBE 的多处理器系统上对维数大于第二阈值、例如 2K 的较大型矩阵进行当前工作条带的 QR 分解。

具体地，本实施例的第二方案，在一次迭代中，如图 10 所示，首先在步骤 100，将上述未处理的矩阵部分划分为 $m \times n$ 块，其中每一个块的大小为 $N_b \times N_b$。在本实施例中，该 $N_b \times N_b$ 例如是 64×64。

也就是说，与现有的条带 QR 分解方案相比，在本实施例的第二方案中，增大了矩阵的块大小。其原因在于，增加块的大小能够降低系统主存与各加速器之间的存储器带宽需求。因为，如上所述，在块大小为 32×32 的情况下，利用 8 个 SPU 进行 QR 分解运算的存储器带宽需求将是 20.6GB/秒。而在块大小为 64×64 的情况下，存储器带宽需求将降低至 18.4GB/秒，这对于能够维持大致 20.5GB/秒存储器带宽的 CBE 这样的多处理器系统而言，是完全能够承受的。

接着，在进行块划分之后，在本实施例的第二方案中，根据

$$A^{(k)} = \left(A_1^{(k)} \quad A_2^{(k)}\right) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = Q \cdot \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

执行以下步骤 101~103。

在步骤 101，参照图 6，在未处理矩阵部分 $A^{(k)}$ 的当前工作条带 $A_1^{(k)}$ 上进行 QR 分解，以将其分解为 V 部分和 R 部分。

具体地，在步骤 101 的子步骤 101-1，从上述未处理矩阵部分 $A^{(k)}$ 划分出由 $m \times n_b$ 块构成的条带 $A_1^{(k)}$，作为当前工作条带，并如图 11(a)所示，进而将该当前工作条带 $A_1^{(k)}$ 划分为大小相等的左右两个子条带 $A_{11}^{(k)}$ 和 $A_{12}^{(k)}$，并且如图 11(b)所示，协调上述多个加速器在左侧子条带 $A_{11}^{(k)}$ 上进行 QR 分解计算。

在步骤 101 的子步骤 101-2，根据步骤 101-1 的计算结果，协调上述多个加速器计算上述左侧子条带 $A_{11}^{(k)}$的三角因子 T。

在步骤 101 的子步骤 101-3，将上述左侧子条带 $A_{11}^{(k)}$以及 $A_{11}^{(k)}$的三角因子 T 应用于右侧子条带 $A_{12}^{(k)}$，以更新该右侧子条带 $A_{12}^{(k)}$的数据。

在步骤 101 的子步骤 101-4，协调上述多个加速器在更新后的上述右侧子条带 $A_{12}^{(k)}$上进行 QR 分解计算。

在步骤 102，根据步骤 101 的计算结果，协调上述多个加速器计算当前工作条带 $A_1^{(k)}$的三角因子 T。

在步骤 103，将当前工作条带 $A_1^{(k)}$以及 $A_1^{(k)}$的三角因子 T 应用于 $A^{(k)}$的剩余的矩阵部分 $A_2^{(k)}$，以更新该矩阵部分 $A_2^{(k)}$的数据。并且，在更新后的矩阵中，如图 11(b)所示，深灰色的矩阵部分 $\widetilde{A}_2^{(k)}$成为第 k+1 次迭代运算的对象。

也就是说，在本发明的第二方案中，将在现有的条带 QR 分解方案中作为第 1 步 QR 分解计算的对象的当前工作条带 $A_1^{(k)}$进一步分为多个子条带，并分别在各个子条带上进行 QR 分解计算。

下面仍以 3×3 块的矩阵（只有一次迭代的情况）为例，说明本实施例的上述第二方案的过程。参照图 12(a)，本实施例的该第二方案将现有的条带 QR 分解方案的第 1 步进而分为 4 个子步骤，其中将作为该第 1 步 QR 分解计算的对象的左侧 3×1 块的条带进一步划分为左右两个子条带，并首先在左侧的子条带上进行 QR 分解计算，并用计算结果更新右侧的子条带，然后在更新后的右侧子条带上进行 QR 分解计算。然后，如图 12(b)、(c)所示，与现有的条带 QR 分解方案相同，在第 2 步计算当前工作条带的三角因子 T_k，在第 3 步将当前工作条带及其三角因子 T_k应用于剩余的矩阵部分以更新其数据。

以上就是对本实施例的在多处理器系统上对矩阵进行 QR 分解的第二方案的详细描述。在该第二方案中，通过增加块的大小，能够降低诸如 SPU 的加速器与主存储器之间的存储器带宽需求，并且通过在一次迭代中将作为 QR 分解计算的对象的当前工作条带进一步划分为多个子条带并分别在

这多个子条带上进行QR分解计算，能够降低因块大小的增加而带来的QR分解计算的复杂度、进而计算时间的增加。

需要说明的是，在本实施例的上述第二方案中，在步骤101和102中也涉及到从该多处理器系统的主存储器到上述多个加速器的矩阵数据的分发，对此，也可以采用图8所示出的矩阵划分方式。

此外，还需要说明的是，在本实施例的上述第二方案中，虽然将当前工作条带 $A_1^{(k)}$ 进一步划分为左右两个子条带 $A_{11}^{(k)}$ 和 $A_{12}^{(k)}$ ，但是，并不限于此，也可以将其划分为更多的子条带。并且，在上述实施例中，虽然将当前工作条带 $A_1^{(k)}$ 划分为大小相等的左右两个子条带 $A_{11}^{(k)}$ 和 $A_{12}^{(k)}$ ，但是，并不限于此，也可以划分为大小不等的子条带。

还需要说明的是，虽然在图5所示出的方法中，根据未处理矩阵的维数大小采取不同的QR分解方案，但是，也可以不判断未处理矩阵的维数而在任何情况下都采取图10和11所示出的第二方案。

在同一发明构思下，本发明提供一种在多处理器系统中对矩阵进行QR分解的装置。下面结合附图对其进行描述。

图13是根据本发明实施例的在多处理器系统中对矩阵进行QR分解的装置的方框图。其中，该多处理器系统具有至少一个核心处理器以及多个加速器。具体地，该多处理器系统例如可以是前述具有一个PPU（核心处理器）和8个SPU（加速器）的CBE。

本实施例的在多处理器系统中对矩阵进行QR分解的装置，对于给定的M×N矩阵A，通过迭代地每次在矩阵的一个条带上进行分解运算来最终将该矩阵A分解为M×M正交矩阵Q和M×N上三角矩阵R的积。

具体地，如图13所示，本实施例的在多处理器系统中对矩阵进行QR分解的装置13包括：选择单元131、常规QR分解单元132、第一方案模块133、第二方案模块134。

其中，选择单元131在每一次迭代中判断上述矩阵A中未处理的矩阵部分（输入矩阵）的维数是否小于第一阈值，如果是，则针对该未处理的矩阵部分启动常规QR分解单元132；否则，判断该未处理的矩阵部分的

维数是否大于第一阈值而小于第二阈值，如果是，则针对该未处理的矩阵部分启动第一方案模块 133；否则，启动第二方案模块 134。

优选地，上述第一阈值是根据上述多处理器系统中的多个加速器之间的通信带宽的大小而确定的，其例如可以是 256；第二阈值是根据上述多个加速器的本地存储器的容量总计而确定的，其例如可以是 2K。

常规 QR 分解单元 132，针对维数小于第一阈值的上述未处理的矩阵部分，按照第一预定的块大小、例如 32 × 32，将其划分为多个块，并仅启用核心处理器对其中包括多个块的当前工作条带进行 QR 分解运算。即该常规 QR 分解单元 132 是根据前述现有的条带 QR 分解方案来实现的。

第一方案模块 113，对于维数大于第一阈值而小于第二阈值的上述未处理的矩阵部分，采用第一方案对其进行 QR 分解运算。

如图 13 所示，该第一方案模块 133 可以进一步包括：块划分单元 1331，用于按照预定的块大小、例如 32 × 32，将上述未处理的矩阵部分划分为多个块；数据分发单元 1332，用于将对上述未处理的矩阵部分中、包括多个块的当前工作条带进行 QR 分解运算所需的矩阵数据从上述多处理器系统的主存储器全部分发给上述多个加速器；判断单元 1333，用于为上述多个加速器的每一个，判断其进行计算所需的数据是否存在于该加速器本地；数据获取单元 1334，用于为上述多个加速器中进行计算所需的数据不存在于该加速器本地的加速器，搜索其他加速器以获取该所需的数据；以及 QR 分解单元 1335，用于协调上述多个加速器利用从其本地或其他的加速器获取的数据进行上述当前工作条带的 QR 分解、三角因子的计算，并根据计算结果更新当前工作条带之外的矩阵部分。

第二方案模块 134，对于维数大于第二阈值的上述未处理的矩阵部分，采用第二方案对其进行 QR 分解运算。

图 14 是根据本发明实施例的在多处理器系统中对矩阵进行QR分解的第二方案模块的方框图。

如图 14 所示，该第二方案模块 134 包括：块划分单元 1341、条带划分单元 1342、子条带处理单元 1343、三角因子计算单元 1344、矩阵更新

单元 1345。

块划分单元 1341 按照预定的块大小、例如 64×64，将上述未处理的矩阵部分划分为多个块。

条带划分单元 1342 将上述未处理的矩阵部分中、包括多个块的当前处理的条带划分为至少两个子条带。具体地，该条带划分单元 1342 可以将上述当前处理的条带进一步划分为左侧子条带和右侧子条带。

子条带处理单元 1343 利用上述多个加速器逐一在上述至少两个子条带上进行 QR 分解，并利用分解结果更新该至少两个子条带中还未进行 QR 分解的子条带的数据。

在上述当前处理的条带被划分为左侧子条带和右侧子条带的情况下，子条带处理单元 1343 可以进一步包括：子条带 QR 分解单元 13431，用于利用上述多个加速器依次在上述左侧子条带和上述右侧子条带上进行 QR 分解运算；子条带三角因子计算单元 13432，用于在上述左侧子条带的 QR 分解运算结束后，根据该分解运算的结果，计算该左侧子条带的三角因子；以及子条带更新单元 13433，用于利用该左侧子条带及其三角因子更新上述右侧子条带的数据；其中上述子条带 QR 分解单元 13431 在更新后的上述右侧子条带上进行 QR 分解运算。

三角因子计算单元 1344，在上述至少两个子条带均进行了 QR 分解运算之后，计算作为该至少两个子条带的整体的、上述当前处理的条带的三角因子。

矩阵更新单元 1345，利用上述当前处理的条带及其三角因子更新上述矩阵中还未进行迭代运算的部分的数据。

以上就是对本实施例的在多处理器系统中对矩阵进行 QR 分解的装置的详细描述。其中，该装置 13 及其各个组成部分，可以由专用的电路或芯片构成，也可以通过计算机（处理器）执行相应的程序来实现。

需要说明的是，虽然在图 13 所示出的装置 13 中，根据未处理矩阵的维数大小启动不同的 QR 分解模块，但是，也可以不判断未处理矩阵的维数而在任何情况下都启动第二方案模块 134。

以上虽然通过一些示例性的实施例对本发明的在多处理器系统上对矩阵进行 QR 分解的方法和装置进行了详细的描述，但是以上这些实施例并不是穷举的，本领域技术人员可以在本发明的精神和范围内实现各种变化和修改。因此，本发明并不限于这些实施例，本发明的范围仅以所附权利要求为准。

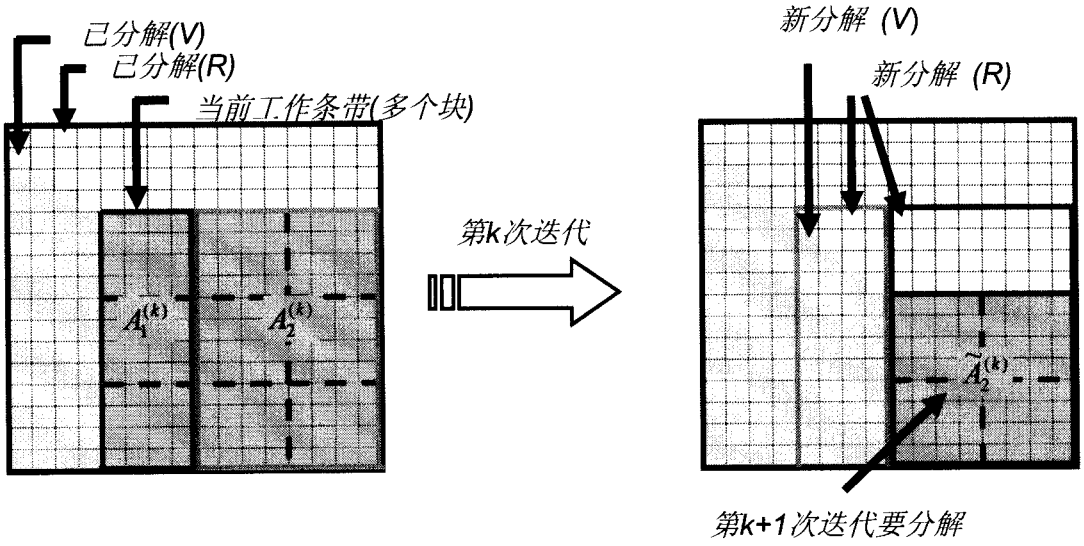

已分解(V)
已分解(R)
当前工作条带(多个块)
第k次迭代
新分解 (V)
新分解 (R)
第k+1次迭代要分解

图 1(a)

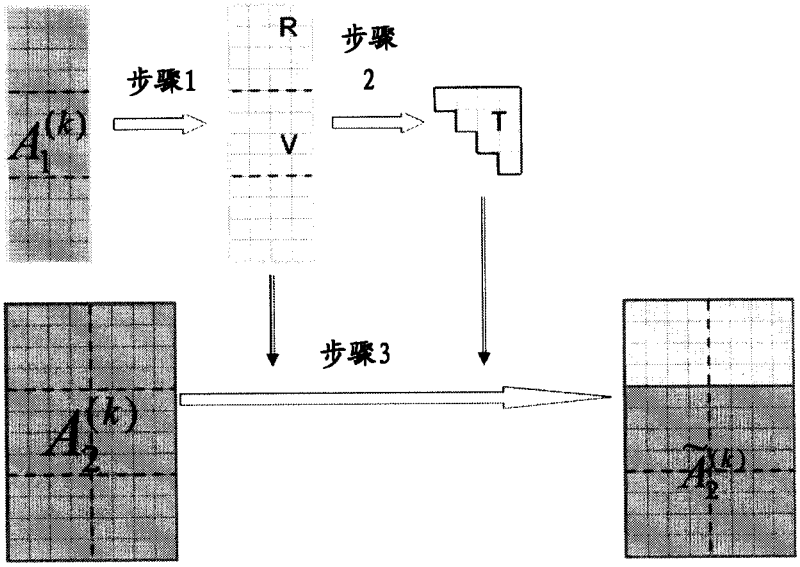

步骤1
R
V
步骤2
T
步骤3

图 1(b)

for $k=1;\ k \le \min(m,n)/n_b;\ k++$
do
1. **DGEQR2**: Compute the QR factorization on an $m' * n_b$ panel of A
   for $i=1;\ i \le n_b;\ i++$
   do
   DLARFG: generate the elementary reflector v_i and τ_i
   DLARF: update the trailing submatrix
   end
2. **DLARFT**: Compute the triangular factor T of the block reflector $Q=(I-VTV^T)$;
3. **DLARFB**: Apply Q^T to the rest of the matrix from the left

$$\begin{pmatrix} R_{12} \\ \tilde{A}_{22} \end{pmatrix} = Q^T \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} = (I - VT^T V^T) \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix}$$

DGEMM: $W = V^T \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix}$

DTRMM: $W = T^T W$

DGEMM: $\begin{pmatrix} R_{12} \\ \tilde{A}_{22} \end{pmatrix} = \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} - VW$

end

图 1(c)

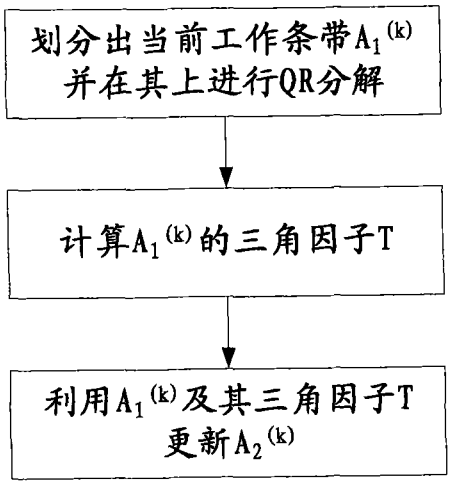


图2

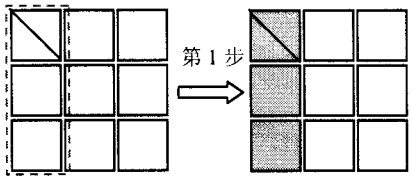


图 3(a)

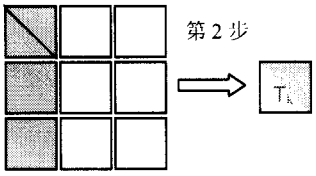


图 3(b)

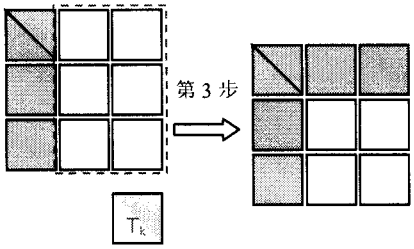


图 3(c)

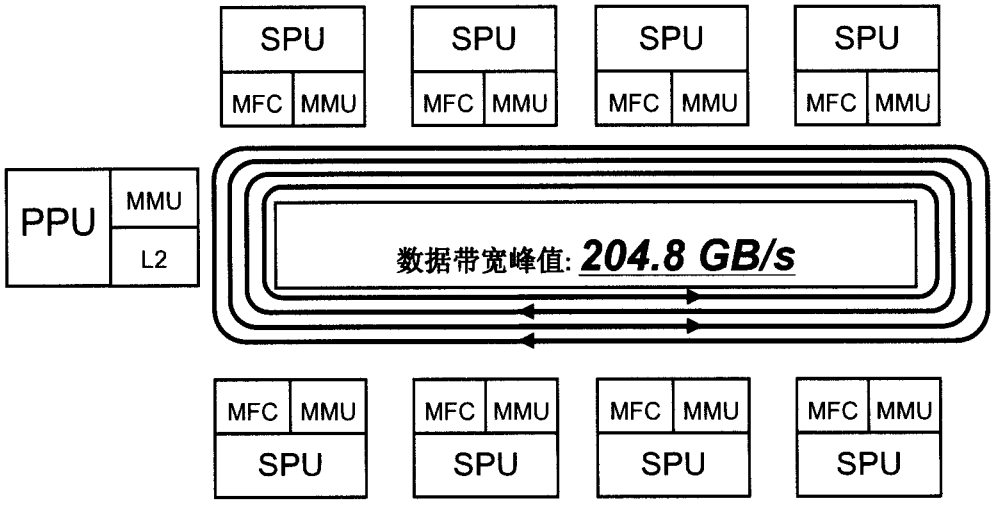


图 4

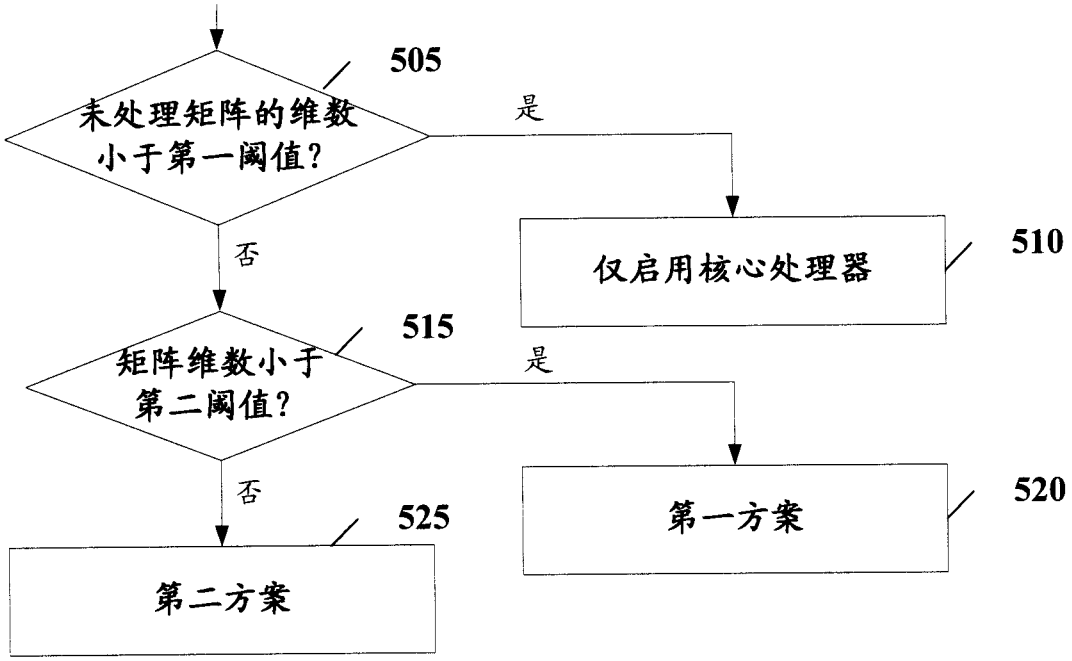

505
未处理矩阵的维数小于第一阈值?
是
否
仅启用核心处理器
510
515
矩阵维数小于第二阈值?
是
否
第一方案
520
525
第二方案

图5

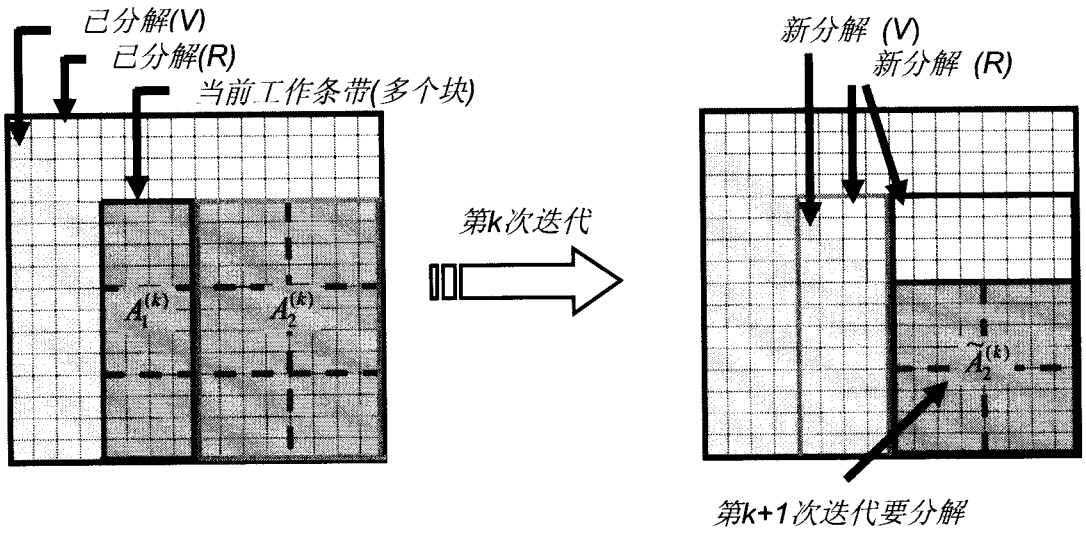

已分解(V)
已分解(R)
当前工作条带(多个块)
第k次迭代
新分解 (V)
新分解 (R)
第k+1次迭代要分解

图 6

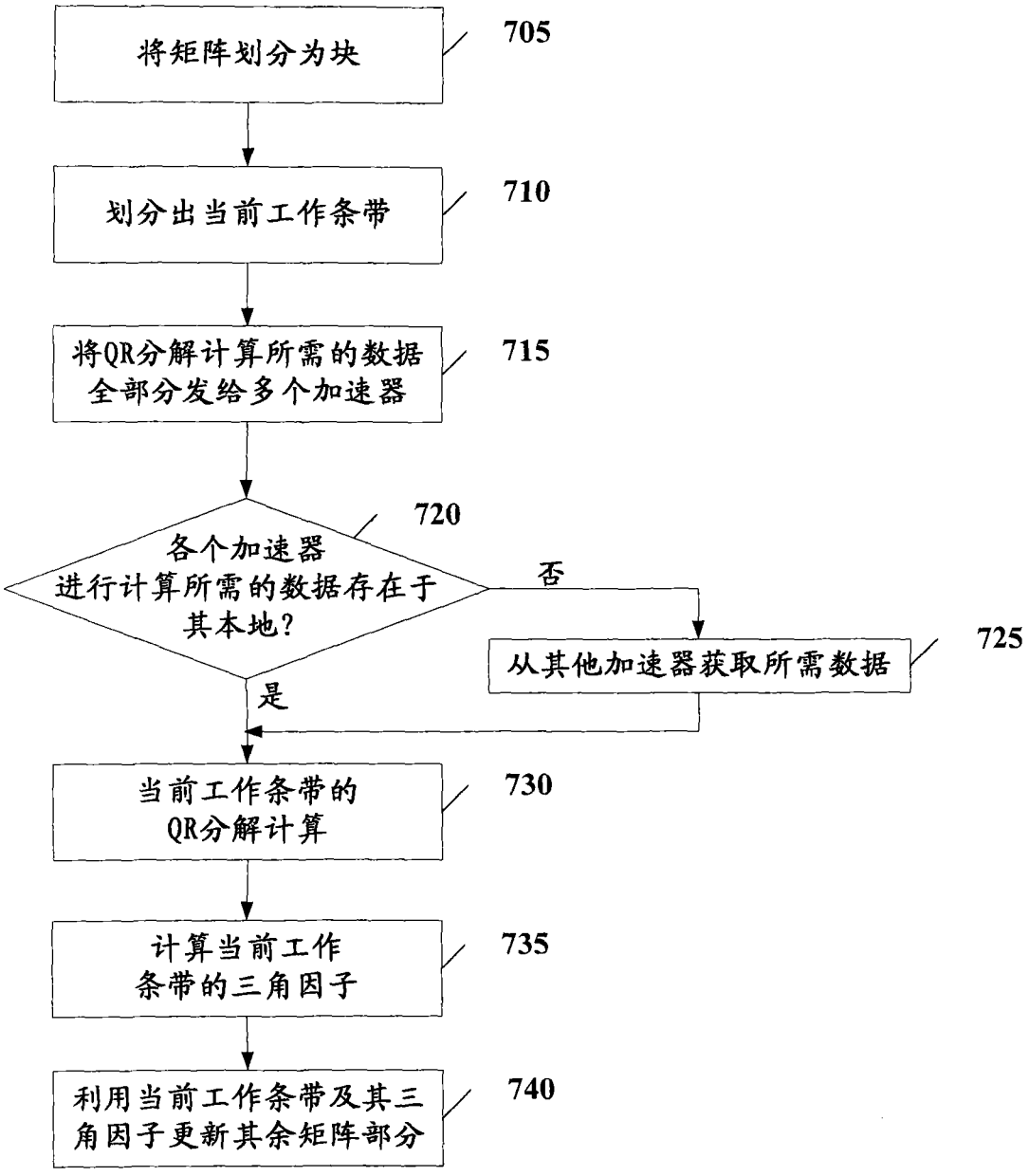

将矩阵划分为块
705
划分出当前工作条带
710
将QR分解计算所需的数据全部分发给多个加速器
715
各个加速器进行计算所需的数据存在于其本地?
720
否
从其他加速器获取所需数据
725
是
当前工作条带的QR分解计算
730
计算当前工作条带的三角因子
735
利用当前工作条带及其三角因子更新其余矩阵部分
740

图7

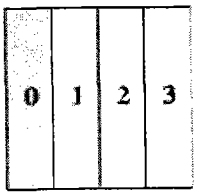


图 8(a)

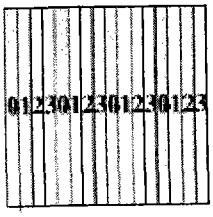


图 8(b)

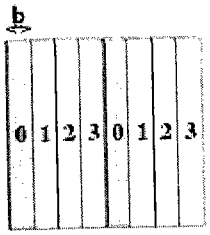


图 8(c)

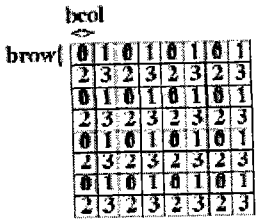


图 8(d)

0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

图 8(e)

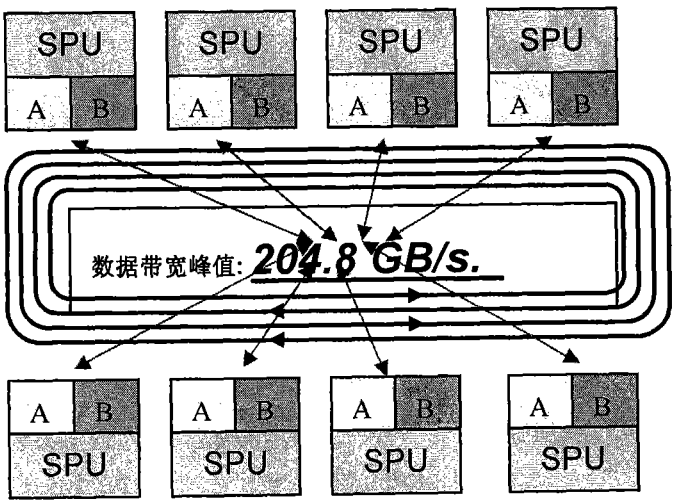


图 9

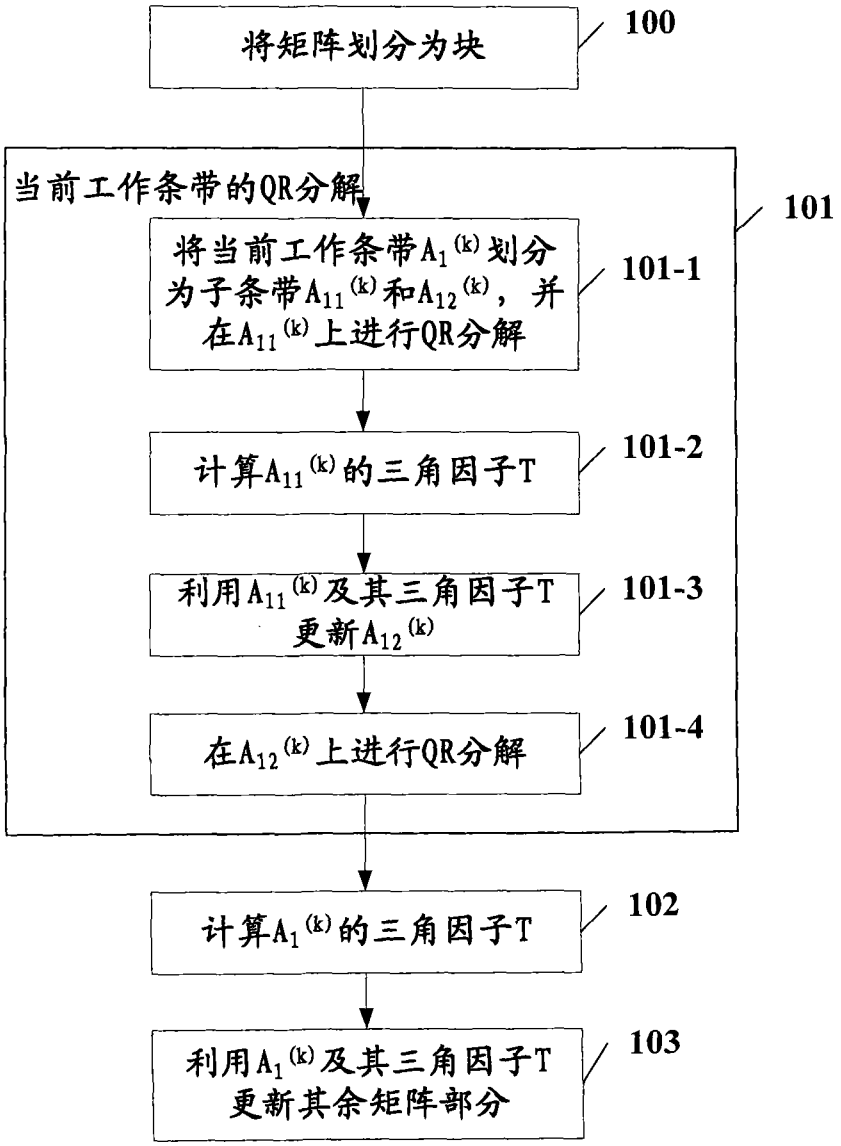

将矩阵划分为块
100
当前工作条带的QR分解
101
将当前工作条带$A_1^{(k)}$划分为子条带$A_{11}^{(k)}$和$A_{12}^{(k)}$，并在$A_{11}^{(k)}$上进行QR分解
101-1
计算$A_{11}^{(k)}$的三角因子T
101-2
利用$A_{11}^{(k)}$及其三角因子T更新$A_{12}^{(k)}$
101-3
在$A_{12}^{(k)}$上进行QR分解
101-4
计算$A_1^{(k)}$的三角因子T
102
利用$A_1^{(k)}$及其三角因子T更新其余矩阵部分
103

图10

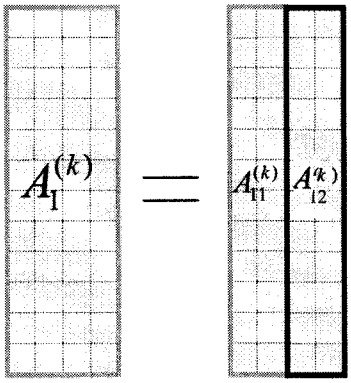


图 11(a)

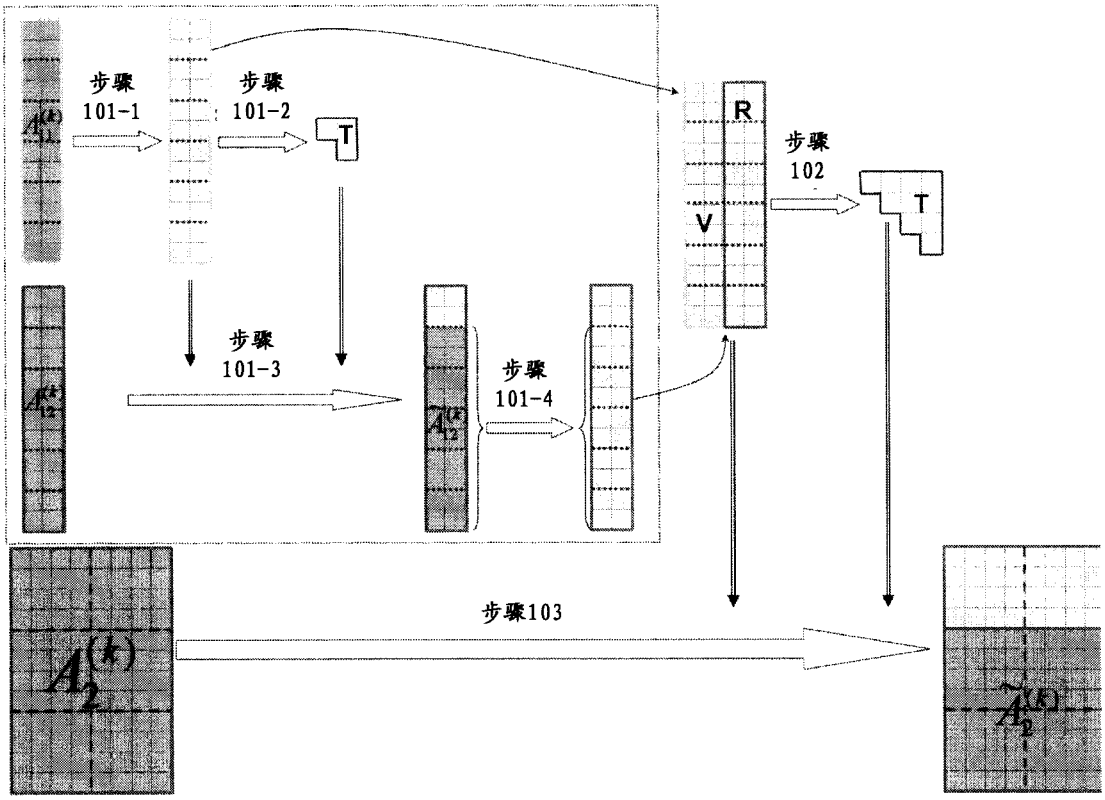


图11(b)

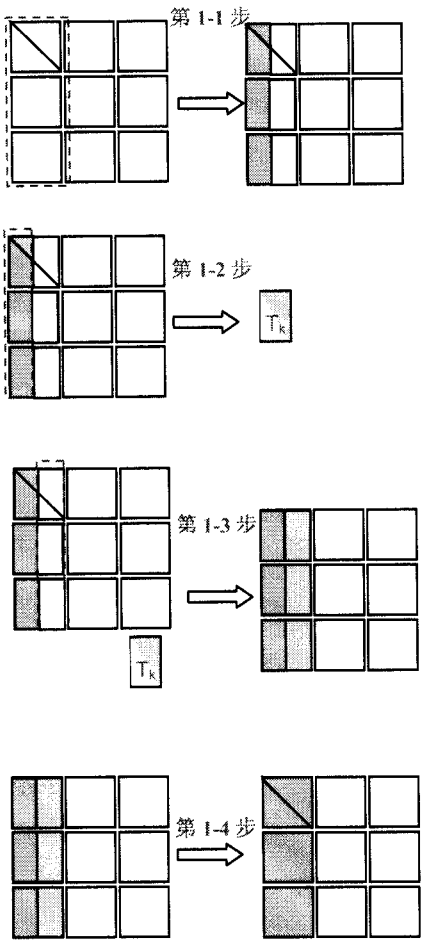


图 12(a)

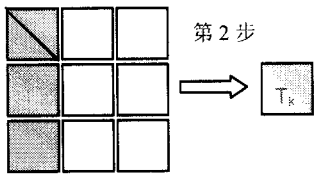


图 12(b)

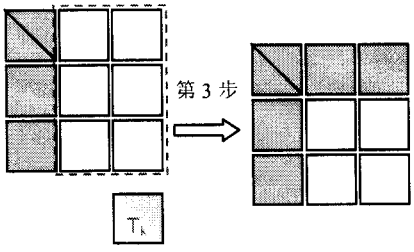


图 12(c)

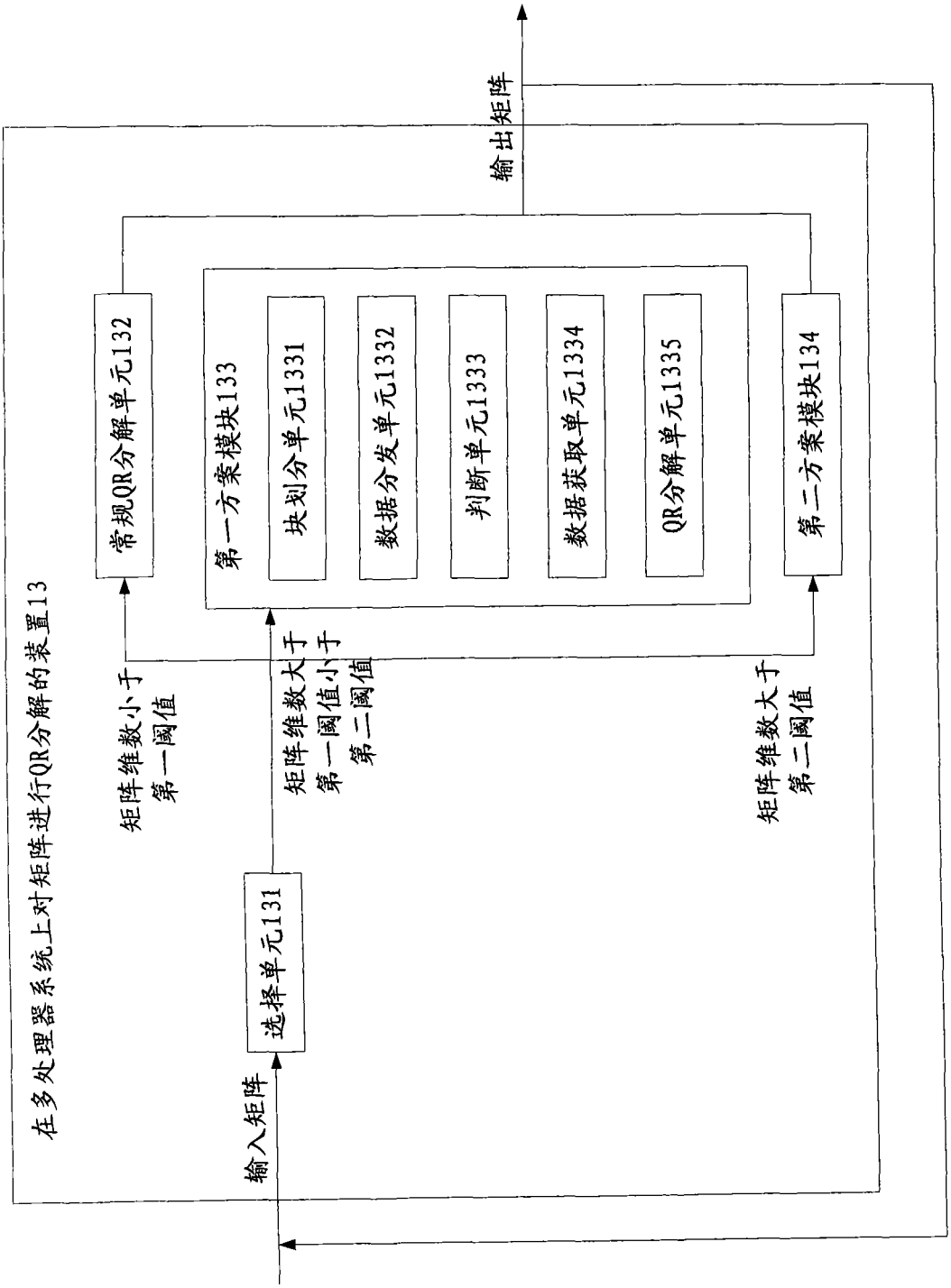

在多处理器系统上对矩阵进行QR分解的装置13
输入矩阵
选择单元131
矩阵维数小于第一阈值
矩阵维数大于第一阈值小于第二阈值
矩阵维数大于第二阈值
常规QR分解单元132
第一方案模块133
块划分单元1331
数据分发单元1332
判断单元1333
数据获取单元1334
QR分解单元1335
第二方案模块134
输出矩阵

图13

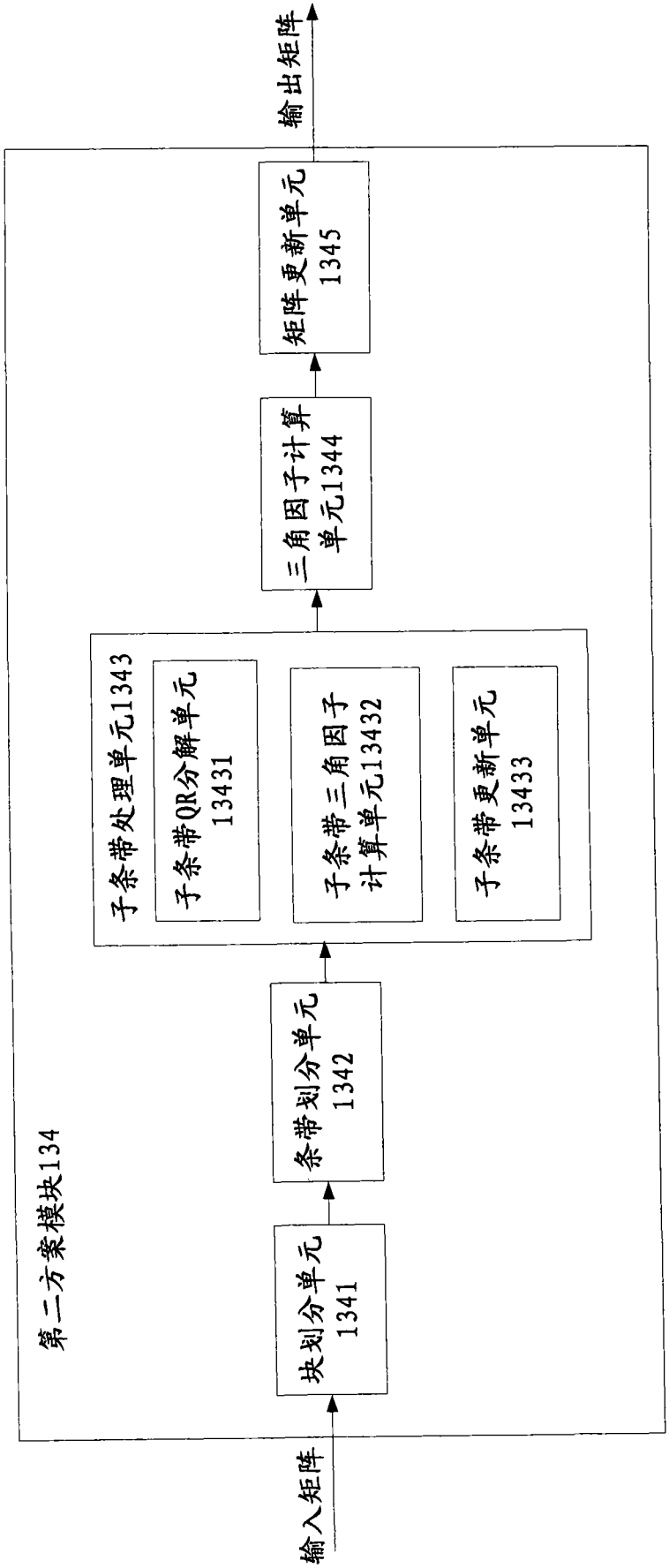

输入矩阵
第二方案模块134
块划分单元
1341
条带划分单元
1342
子条带处理单元1343
子条带QR分解单元
13431
子条带三角因子
计算单元13432
子条带更新单元
13433
三角因子计算
单元1344
矩阵更新单元
1345
输出矩阵

图14