(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103902507 B
(45)授权公告日 2017.05.10

(21)申请号 201410123578.6

(22)申请日 2014.03.28

(65)同一申请的已公布的文献号
申请公布号 CN 103902507 A

(43)申请公布日 2014.07.02

(73)专利权人 中国科学院自动化研究所
地址 100190 北京市海淀区中关村东路95号

(72)发明人 郭晓龙 王晓琴 王伟康 吴军宁 林啸 郭璟 张森 赵旭莹

(74)专利代理机构 中科专利商标代理有限责任公司 11021
代理人 宋焰琴

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

(56)对比文件
CN 103236903 A,2013.08.07,
CN 102541749 A,2012.07.04,
CN 102541774 A,2012.07.04,
张凯等.基于 SIMD处理器的全定制多粒度矩阵寄存器文件.《国防科技大学学报》.2013,第35卷(第4期),第156-160页.

审查员 李欢欢

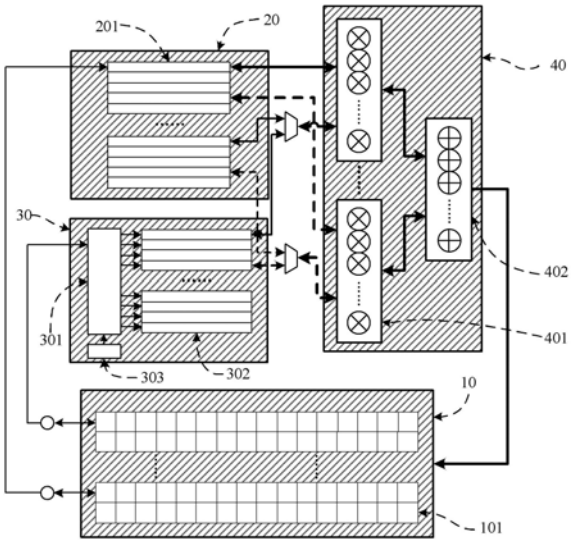
权利要求书2页 说明书7页 附图4页

(54)发明名称

一种面向可编程代数处理器的矩阵乘法计算装置及方法

(57)摘要

本发明公开了一种矩阵乘法计算装置及方法,所述装置包括多粒度并行存储器、数据缓存装置、数据广播缓存装置和向量运算装置。本发明采用可编程配置的DSP芯片,并结合高效的向量化矩阵乘法方案,针对实际应用中存在的矩阵尺寸小、运算量大的矩阵乘法进行并行优化处理,具有运算速度快,并行粒度高和访存次数少的优点。



CN 103902507 B

1.一种矩阵乘法计算装置,其特征在于,该装置包括多粒度并行存储器(10)、数据缓存装置(20)、数据广播缓存装置(30)和向量运算装置(40);

所述的多粒度并行存储器(10)用于存储要进行乘法运算的矩阵、广播索引以及矩阵乘法之后的结果;

所述数据缓存装置(20)用于暂存从多粒度并行存储器(10)中取出的要进行乘法运算的矩阵;

所述数据广播缓存装置(30)用于将要进行数据广播操作的矩阵从多粒度并行存储器(10)中取出,并对所述矩阵的数据进行广播操作;所述数据广播缓存装置(30)包括数据广播控制单元(301)、数据缓存实体(302)和广播索引寄存器(303),其中数据广播控制单元(301)用于控制数据广播操作;

所述向量运算装置(40)用于将从所述数据缓存装置(20)中读取得到的矩阵进行向量运算,或将从所述数据缓存装置(20)中读取得到的矩阵和从所述数据广播缓存装置(30)中读取的矩阵进行向量运算,并将结果写入所述多粒度存储器(10)中。

2.如权利要求1所述的矩阵乘法计算装置,其特征在于,所述多粒度并行存储器的读写位宽、数据缓存装置(20)中寄存器堆(201)的位宽、数据广播缓存装置中相关寄存器位宽以及所述向量运算装置(40)的运算尺寸相等。

3.如权利要求1所述的矩阵乘法计算装置,其特征在于,所述数据广播缓存装置对所述矩阵的数据依据广播索引寄存器(303)中的广播索引进行广播操作。

4.如权利要求1所述的矩阵乘法计算装置,其特征在于,所述向量运算单元(40)包括乘法运算单元(401)和累加运算单元(402)。

5.一种矩阵乘法计算方法,其特征在于,包括如下步骤:

步骤S1:分别从多粒度并行存储器10中按行读取L×M行的A系列矩阵以及按行读取M×N行的B系列矩阵到数据缓存装置(20)中,A系列矩阵放置在寄存器堆(201)中的C_k寄存器中,B系列矩阵放置在寄存器堆(201)中的寄存器中D_l,其中k∈[1,L*M],l∈[1,M*N];

步骤S2:令k1=0,k2=0;

步骤S3:取C_k中第k1×M+1到(k1+1)×M行数据和D_l中第k2×M+1到(k2+1)×M行数据,其相应行分别进行点乘操作,然后将结果进行累加操作,得到结果E,最后将E写回到多粒度并行存储器(10)中;

步骤S4:k2加1,重复步骤S3,直到k2等于N为止;

步骤S5:k1加1,重复步骤S3~S4,直到k1等于L为止;

步骤S6:读取下一个L×M行的A系列矩阵和M×N行的B系列矩阵到数据缓存装置(20)中,重复步骤S2~S5,直到所有矩阵计算完毕。

6.一种矩阵乘法计算方法,其特征在于,包括如下步骤:

步骤P1:从多粒度并行存储器(10)中按列读取L×M行1列A矩阵数据到数据广播缓存装置(30)中,并且按行读取M×N行B系列矩阵数据到数据缓存装置(20)中,表示为D_l,其中l∈[1,M*N];

步骤P2:对数据广播缓存装置(30)中的每一个数据进行广播操作,即每一个数据都复制BS份存储在寄存器C_k中,其中k∈[1,L*M],BS为存储器端口位宽所能容纳的最大数据个数;

步骤P3:令k1＝0,k2＝0;

步骤P4:取C_k中第k1×M+1到(k1+1)×M行数据和D_1中第k2×M+1到(k2+1)×M行数据,其相应行分别进行点乘操作,然后将结果进行累加操作,得到结果E,最后将E写回到多粒度并行存储器(10)中;

步骤P5:k2加1,重复步骤P4,直到k2等于N为止;

步骤P6:k1加1,重复步骤P4～P5,直到k1等于L为止;

步骤P7:读取下一个M×N行的B系列矩阵到数据缓存装置(20)中,重复步骤P3～P6,直到所有矩阵计算完毕。

7.一种矩阵乘法计算方法,其特征在于,包括如下步骤:

步骤Q1:从多粒度并行存储器(10)中按行读取L×M行A系列矩阵数据到数据缓存装置(20)中,表示为C_k,其中k∈[1,L×M],并且按列读取M×N行1列B系列矩阵数据到数据广播缓存装置(30)中;

步骤Q2:对数据广播缓存装置(30)中的每一个数据进行广播操作,即每一个数据都复制BS份存储在寄存器D_1中,其中1∈[1,M×N],BS为存储器端口位宽所能容纳的最大数据个数;

步骤Q3:令k1＝0,k2＝0;

步骤Q4:取C_k中第k1×M+1到(k1+1)×M行数据和D_1中第k2×M+1到(k2+1)×M行数据,其相应行分别进行点乘操作,然后将结果进行累加操作,得到结果E,最后将E写回到多粒度并行存储器(10)中;

步骤Q5:k2加1,重复步骤Q4,直到k2等于N为止;

步骤Q6:k1加1,重复步骤Q4～Q5,直到k1等于L为止;

步骤Q7:读取下一个L×M行的A系列矩阵到数据缓存装置(20)中,重复步骤Q3～Q6,直到所有矩阵计算完毕。

8.如权利要求5至7中任一项所述的矩阵乘法计算方法,其特征在于,所述A系列矩阵以行优先存储在多粒度并行存储器(10)的一列Bank内;所述B系列矩阵以列优先存储在多粒度并行存储器(10)的一列Bank内。

一种面向可编程代数处理器的矩阵乘法计算装置及方法

技术领域

[0001] 本发明涉及数据处理技术领域，更具体地，涉及一种基于可编程代数处理器的矩阵乘法计算装置及方法。

背景技术

[0002] 矩阵乘法是科学计算中一种基本操作，其广泛用于信号处理、图像处理、雷达、声纳、通信等复杂计算领域中，并且由于其计算复杂度为$O(n^3)$，使得矩阵乘法往往成为算法计算过程中最为耗时的操作，进而影响整个算法的性能。矩阵乘法操作又分为大矩阵乘法和小矩阵乘法，大矩阵乘法由于其行列数值非常巨大，导致运算量呈指数级增加，近些年受到广泛关注，一种通用的处理方法为将大矩阵进行分块处理，以加快其运算效率；小矩阵乘法由于其单个乘法的计算量并不是很大，往往被人们所忽视，但随着无线通信领域、雷达信号处理领域，数字图像处理等计算密集型领域的广泛发展，海量信息必须在固定时间内进行处理，而其中大规模的小矩阵乘法随着矩阵数量的急剧增加，逐渐成为非常耗时的计算操作。

[0003] 例如在无线通信领域中的预编码过程，存在着多种模式的大规模小矩阵乘法，并且其有非常严格的时间约束。具体说来，多天线技术包括传输分集、空间复用和波束赋形技术。这三种技术简单来说都包含大规模小矩阵相乘，以空间复用为例，其分为闭环空间复用和开环空间复用，其计算公式如下：

[0004] 闭环空间复用：

[0005] $$\begin{bmatrix} y^{(0)}(i) \\ \vdots \\ y^{(P-1)}(i) \end{bmatrix} = W(i) \begin{bmatrix} x^{(0)}(i) \\ \vdots \\ x^{(\upsilon-1)}(i) \end{bmatrix}$$

[0006] 开环空间复用：

[0007] $$\begin{bmatrix} y^{(0)}(i) \\ \vdots \\ y^{(P-1)}(i) \end{bmatrix} = W(i)D(i)U \begin{bmatrix} x^{(0)}(i) \\ \vdots \\ x^{(\upsilon-1)}(i) \end{bmatrix}$$

[0008] 其中：

[0009]

Number of layers υ	U	$D(i)$
2	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & e^{-j2\pi/2}\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ 0 & e^{-j2\pi i/2}\end{bmatrix}$
3	$\frac{1}{\sqrt{3}}\begin{bmatrix}1 & 1 & 1\\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3}\\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3}\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0\\ 0 & e^{-j2\pi i/3} & 0\\ 0 & 0 & e^{-j4\pi i/3}\end{bmatrix}$
4	$\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & e^{-j2\pi/4} & e^{-j4\pi/4} & e^{-j6\pi/4}\\ 1 & e^{-j4\pi/4} & e^{-j8\pi/4} & e^{-j12\pi/4}\\ 1 & e^{-j6\pi/4} & e^{-j12\pi/4} & e^{-j18\pi/4}\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & e^{-j2\pi i/4} & 0 & 0\\ 0 & 0 & e^{-j4\pi i/4} & 0\\ 0 & 0 & 0 & e^{-j6\pi i/4}\end{bmatrix}$

[0010] W(i)为码本,根据天线数不同存在着不同的码本,例如,在两天线情况下存在如下码本:

[0011]

Codebook index	Number of layers υ	
	1	2
0	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}$	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
1	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
2	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ j\end{bmatrix}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\ j & -j\end{bmatrix}$
3	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -j\end{bmatrix}$	-

[0012] 四天线情况下具体码本由$W_n = I - 2u_n u_n^H / u_n^H u_n$得出,$u_n$及Wn从下表得出:

[0013]

Codebook index	u_n	Number of layers υ			
		1	2	3	4
0	$u_0 = \begin{bmatrix}1 & -1 & -1 & -1\end{bmatrix}^T$	$W_0^{\{1\}}$	$W_0^{\{14\}}/\sqrt{2}$	$W_0^{\{124\}}/\sqrt{3}$	$W_0^{\{1234\}}/2$
1	$u_1 = \begin{bmatrix}1 & -j & 1 & j\end{bmatrix}^T$	$W_1^{\{1\}}$	$W_1^{\{12\}}/\sqrt{2}$	$W_1^{\{123\}}/\sqrt{3}$	$W_1^{\{1234\}}/2$
2	$u_2 = \begin{bmatrix}1 & 1 & -1 & 1\end{bmatrix}^T$	$W_2^{\{1\}}$	$W_2^{\{12\}}/\sqrt{2}$	$W_2^{\{123\}}/\sqrt{3}$	$W_2^{\{3214\}}/2$
3	$u_3 = \begin{bmatrix}1 & j & 1 & -j\end{bmatrix}^T$	$W_3^{\{1\}}$	$W_3^{\{12\}}/\sqrt{2}$	$W_3^{\{123\}}/\sqrt{3}$	$W_3^{\{3214\}}/2$
4	$u_4 = \begin{bmatrix}1 & (-1-j)/\sqrt{2} & -j & (1-j)/\sqrt{2}\end{bmatrix}^T$	$W_4^{\{1\}}$	$W_4^{\{14\}}/\sqrt{2}$	$W_4^{\{124\}}/\sqrt{3}$	$W_4^{\{1234\}}/2$
5	$u_5 = \begin{bmatrix}1 & (1-j)/\sqrt{2} & j & (-1-j)/\sqrt{2}\end{bmatrix}^T$	$W_5^{\{1\}}$	$W_5^{\{14\}}/\sqrt{2}$	$W_5^{\{124\}}/\sqrt{3}$	$W_5^{\{1234\}}/2$
6	$u_6 = \begin{bmatrix}1 & (1+j)/\sqrt{2} & -j & (-1+j)/\sqrt{2}\end{bmatrix}^T$	$W_6^{\{1\}}$	$W_6^{\{13\}}/\sqrt{2}$	$W_6^{\{134\}}/\sqrt{3}$	$W_6^{\{1324\}}/2$
7	$u_7 = \begin{bmatrix}1 & (-1+j)/\sqrt{2} & j & (1+j)/\sqrt{2}\end{bmatrix}^T$	$W_7^{\{1\}}$	$W_7^{\{13\}}/\sqrt{2}$	$W_7^{\{134\}}/\sqrt{3}$	$W_7^{\{1324\}}/2$

[0014]

8	$u_8=[1\ \ -1\ \ 1\ \ 1]^T$	$W_8^{\{1\}}$	$W_8^{\{12\}}/\sqrt{2}$	$W_8^{\{124\}}/\sqrt{3}$	$W_8^{\{1234\}}/2$
9	$u_9=[1\ \ -j\ \ -1\ \ -j]^T$	$W_9^{\{1\}}$	$W_9^{\{14\}}/\sqrt{2}$	$W_9^{\{134\}}/\sqrt{3}$	$W_9^{\{1234\}}/2$
10	$u_{10}=[1\ \ 1\ \ 1\ \ -1]^T$	$W_{10}^{\{1\}}$	$W_{10}^{\{13\}}/\sqrt{2}$	$W_{10}^{\{123\}}/\sqrt{3}$	$W_{10}^{\{1324\}}/2$
11	$u_{11}=[1\ \ j\ \ -1\ \ j]^T$	$W_{11}^{\{1\}}$	$W_{11}^{\{13\}}/\sqrt{2}$	$W_{11}^{\{134\}}/\sqrt{3}$	$W_{11}^{\{1324\}}/2$
12	$u_{12}=[1\ \ -1\ \ -1\ \ 1]^T$	$W_{12}^{\{1\}}$	$W_{12}^{\{12\}}/\sqrt{2}$	$W_{12}^{\{123\}}/\sqrt{3}$	$W_{12}^{\{1234\}}/2$
13	$u_{13}=[1\ \ -1\ \ 1\ \ -1]^T$	$W_{13}^{\{1\}}$	$W_{13}^{\{13\}}/\sqrt{2}$	$W_{13}^{\{123\}}/\sqrt{3}$	$W_{13}^{\{1324\}}/2$
14	$u_{14}=[1\ \ 1\ \ -1\ \ -1]^T$	$W_{14}^{\{1\}}$	$W_{14}^{\{13\}}/\sqrt{2}$	$W_{14}^{\{123\}}/\sqrt{3}$	$W_{14}^{\{3214\}}/2$
15	$u_{15}=[1\ \ 1\ \ 1\ \ 1]^T$	$W_{15}^{\{1\}}$	$W_{15}^{\{12\}}/\sqrt{2}$	$W_{15}^{\{123\}}/\sqrt{3}$	$W_{15}^{\{1234\}}/2$

[0015] 八天线码本相对更为复杂一些，这里就不一一列举。然后这些小矩阵码本再和每层的信号进行矩阵乘法，最终得出不同天线上的发射信号，由于信号量非常巨大，导致该过程也变得异常费时。基于非码本的预编码操作，同样是小矩阵码本和信号矩阵相乘，只是码本获得方式不同而已。

[0016] 总体来说，对于无线通信领域，特别是LTE/LTE-A中，存在码本矩阵行列为(1，1)，(2，1)，(2，2)，(4，1)，(4，2)，(4，3)，(4，4)，(8，1)，(8，2)，(8，3)，(8，4)，(8，5)，(8，6)，(8，7)，(8，8)15种情况和信号矩阵行列为(1，1)，(2，1)，(3，1)，(4，1)，(5，1)，(6，1)，(7，1)，(8，1)8种情况，并且码本矩阵和信号矩阵相乘的次数非常多。对于该类矩阵乘法由于其矩阵行列比较小，无法使用分块方法进行计算，而直接行列做乘累加操作又相对耗时，因此有必要提出一种高效地解决上述问题的大规模小矩阵相乘的方法与装置。

发明内容

[0017] (一)要解决的技术问题

[0018] 本发明所要解决的技术问题是现有的矩阵相乘方法与装置对于大规模小矩阵的相乘执行效率不高的问题。

[0019] (二)技术方案

[0020] (三)有益效果

[0021] 本发明对矩阵算法进行了优化，通过“数据缓存、广播及流水”机制，结合一个运算模式多样的运算部件，使得一系列小矩阵乘法能够充分的并行执行，能够提高大规模小矩阵的运算效率。

附图说明

[0022] 图1为本发明的矩阵乘法装置图；

[0023] 图2为本发明中A系列矩阵在多粒度并行存储器中的数据分布图；

[0024] 图3为本发明中B系列矩阵在多粒度并行存储器中的数据分布图；

[0025] 图4为本发明实施例的矩阵乘法的计算示意图；

[0026] 图5为本发明中广播操作示意图。

具体实施方式

[0027] 本发明针对现有的情况，提出了一种高效地计算矩阵乘法的方法和装置，特别适合于大规模小矩阵乘法。

[0028] 本发明所提出的大规模小矩阵乘法向量化装置包含：多粒度并行存储器10、数据缓存装置20、数据广播缓存装置30、向量运算装置40。其中：

[0029] 所述多粒度并行存储器10用于存储多个要进行乘法运算的矩阵、广播索引以及矩阵乘法之后的结果。所述广播索引用于对矩阵中的每个数据进行广播操作。该存储器的读写位宽与数据缓存装置的寄存器堆201位宽、数据广播缓存装置中相关寄存器位宽以及所述向量运算装置40的运算尺寸一致，记为P。

[0030] 所述数据缓存装置20用于将矩阵乘法中的不需要广播的矩阵从多粒度并行存储器10中取出，存入到寄存器堆201中。所述数据缓存装置20由寄存器堆201组成。

[0031] 所述数据广播缓存装置30用于将矩阵乘法中需广播的矩阵数据以及广播索引从所述多粒度并行存储器10中取出，分别放入到数据缓存实体302和广播索引寄存器303中，依据广播索引寄存器303中的广播索引对B系列矩阵数据进行广播操作。所述广播操作是指将一个数据依据广播索引复制多份放入到寄存器的相应位置中，如图5所示。所述数据广播装置30包括数据广播控制单元301和数据缓存实体302和广播索引寄存器303，其中数据广播控制单元301用于控制具体的数据广播操作。

[0032] 所述向量运算装置40，用于将从所述数据缓存装置20中读取得到的要进行乘法运算的矩阵和从所述数据广播缓存装置30中读取并广播后的要进行乘法运算的矩阵进行向量运算，并将结果写入多粒度存储单元101中。所述向量运算单元40包含乘法运算单元401和累加运算单元402，可同时执行P个字节的向量运算操作。

[0033] 本发明的另一方面是提出一种用于大规模小矩阵乘法的向量化计算方法，用于进行矩阵乘法$A_i \times B_i$，其中A_i表示维度为L×M的A系列矩阵，B_i表示维度为M×N的B系列矩阵，i表示矩阵的序号，且i为正整数。根据本发明的计算装置的存储器端口位宽和运算尺寸，一次可同时执行BS对矩阵相乘，其中BS为存储器端口位宽所能容纳的最大数据个数。

[0034] 这里需要说明的是，A系列矩阵和B系列矩阵在内存中按下列规则存储。其中A系列矩阵以行优先存储在多粒度并行存储器10的一个Bank内，如图2所示；B系列矩阵以列优先存储在多粒度并行存储器10的一个Bank内，如图3所示。

[0035] 本发明根据A系列矩阵和B系列矩阵的特点，分为三种情况：

[0036] 一、A系列矩列和B系列矩阵个数都不唯一；

[0037] 二、A系列矩阵个数唯一，B系列矩阵个数不唯一；

[0038] 三、A系列矩阵个数不唯一，B系列矩阵个数唯一。

[0039] 第一种情况，本发明的方法包括如下步骤：

[0040] 步骤S1：分别从多粒度并行存储器10中按行读取L×M行的A系列矩阵以及按行读取M×N行的B系列矩阵到数据缓存装置20中，A系列矩阵放置在寄存器堆201中的C_k寄存器中，B系列矩阵放置在寄存器堆201中的寄存器中D_l，其中$k \in [1, L*M]$，$l \in [1, M*N]$；

[0041] 步骤S2：令k1＝0，k2＝0；

[0042] 步骤S3：取C_k中第k1×M+1到(k1+1)×M行数据和D_l中第k2×M+1到(k2+1)×M行数

据，其相应行分别进行点乘操作，然后将结果进行累加操作，得到结果E，最后将E写回到多粒度并行存储器10中，如图4所示；

[0043] 步骤S4：k2加1，重复步骤S3，直到k2等于N为止；

[0044] 步骤S5：k1加1，重复步骤S3～S4，直到k1等于L为止；

[0045] 步骤S6：读取下一个L×M行的A系列矩阵和M×N行的B系列矩阵到数据缓存装置20中，重复步骤S2～S5，直到所有矩阵计算完毕。

[0046] 第二种情况包括如下步骤：

[0047] 步骤P1：从多粒度并行存储器10中按列读取L×M行1列A矩阵数据到数据广播缓存装置30中，并且按行读取M×N行B系列矩阵数据到数据缓存装置20中，表示为D_l，其中$l \in [1,M*N]$；

[0048] 步骤P2：对数据广播缓存装置30中的每一个数据进行类似于图5所示广播操作，即每一个数据都复制BS份存储在寄存器C_k中，其中$k \in [1,L*M]$；

[0049] 步骤P3：令k1＝0，k2＝0；

[0050] 步骤P4：取C_k中第k1×M+1到(k1+1)×M行数据和D_l中第k2×M+1到(k2+1)×M行数据，其相应行分别进行点乘操作，然后将结果进行累加操作，得到结果E，最后将E写回到多粒度并行存储器10中，如图4所示；

[0051] 步骤P5：k2加1，重复步骤P4，直到k2等于N为止；

[0052] 步骤P6：k1加1，重复步骤P4～P5，直到k1等于L为止；

[0053] 步骤P7：读取下一个M×N行的B系列矩阵到数据缓存装置20中，重复步骤P3～P6，直到所有矩阵计算完毕。

[0054] 第三种情况包括如下步骤：

[0055] 步骤Q1：从多粒度并行存储器10中按行读取L×M行A系列矩阵数据到数据缓存装置20中，表示为C_k，其中$k \in [1,L \times M]$。并且按列读取M×N行1列B系列矩阵数据到数据广播缓存装置30中；

[0056] 步骤Q2：对数据广播缓存装置30中的每一个数据进行类似于图5所示广播操作，即每一个数据都复制BS份存储在寄存器D_l中，其中$l \in [1,M \times N]$；

[0057] 步骤Q3：令k1＝0，k2＝0；

[0058] 步骤Q4：取C_k中第k1×M+1到(k1+1)×M行数据和D_l中第k2×M+1到(k2+1)×M行数据，其相应行分别进行点乘操作，然后将结果进行累加操作，得到结果E，最后将E写回到多粒度并行存储器10中，如图4所示；

[0059] 步骤Q5：k2加1，重复步骤Q4，直到k2等于N为止；

[0060] 步骤Q6：k1加1，重复步骤Q4～Q5，直到k1等于L为止；

[0061] 步骤Q7：读取下一个L×M行的A系列矩阵到数据缓存装置20中，重复步骤Q3～Q6，直到所有矩阵计算完毕。

[0062] 由于本发明的向量运算部件包含多套向量运算单元，并且上述步骤运算操作相对固定，因此本发明可将如上步骤进行流水操作，最大限度的提高计算效率。

[0063] 本发明通过“数据缓存、广播及流水”机制，结合一个运算模式多样的运算部件使得可编程代数处理器可以最大限度的加快小矩阵乘法。极大的提高了硬件利用效率和矩阵运算速率。

[0064] 采用本发明进行大规模小矩阵的矩阵相乘计算，具有以下有益效果：

[0065] 一、运算速度快，由于采用了运算尺寸为BS个数据的向量运算部件，使得每次可以并行执行BS个小矩阵相乘操作。其速率是一般串行速度的BS倍，并且通过采用广播机制，改变矩阵乘法运算模式，减少了矩阵乘法的运算周期数，另外由于运算操作相对固定，可以最大限度的全流水执行，进一步提升了运算速率。

[0066] 二、减少访存次数。由于每次访存时采用的多粒度并行存储器10和向量运算部件30的运算尺寸为BS，使得访存次数减少了近BS倍。并且数据缓存装置20和数据广播缓存装置30可以暂存矩阵乘法过程中的中间变量，使得进一步减少了访存次数，降低了功耗，提高了数据的使用效率。

[0067] 三、适用范围广，由于本装置的运算尺寸非常宽，并不需要固定矩阵尺寸，通过编程配置即可支持多种不同行列宽度的小矩阵的向量化乘法操作，从而扩展了本发明的适用范围。

[0068] 本发明需要利用支持多粒度并行读写的存储器，对于该存储器的描述可参考申请号为201110459453.7的专利申请《多粒度并行存储系统》和申请号为201110460585.1的专利申请《多粒度并行存储系统与存储器》。

[0069] 本发明需要利用交织网络技术，对于该交织网络的描述可参考申请号为201310138909.9的专利申请《一种改变数据序列顺序的装置》。

[0070] 为了使本发明的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本发明进一步详细说明。

[0071] 在本说明书中，为了描述方便，首先介绍一下本发明中所使用到的各个部件的功能说明。

[0072] 多粒度并行存储器

[0073] 本发明中所使用的多粒度并行存储器的数据位宽是以存储单元为单位进行度量的，存储单元定义为存储器的编制单位，也是存储器可读写的最小数据位宽。本发明中均假定最小数据位宽即存储单元为8bit。而“粒度”是指地址连续的存储单元的个数。多粒度指的是可同时读取多个连续的存储单元的数据，最大为存储器的读写端口位宽，假设本发明中所使用数据为单精度浮点数据，一个浮点数据需要4个连续的存储单元来存储。本发明中多粒度并行存储器可一次读写P个存储单元，即一次最大可同时读取BS个数据(P＝BS×4)。另外，本并行存储器支持支持按行或按列读取。

[0074] 数据缓存装置

[0075] 数据缓存装置用于暂存源数据、结果和一些常用的中间数据，减少反复读取存储器的次数。该功能通过一个寄存器堆来实现，该寄存器堆中每个寄存器的位宽为P×8bit。

[0076] 数据广播缓存装置

[0077] 除了具有暂存源数据、结果和一些常用的中间数据的功能外，还可以通过配置广播寄存器对数据进行广播操作。其中广播操作是以字节为单位进行操作的，因此对于单精度浮点数据来说，利用4个字节的广播索引位来控制一个数据的广播位置。

[0078] 向量运算装置

[0079] 向量运算装置主要实现各种计算操作，包括加法、乘法、数据读写等操作，其运算尺寸与多粒度并行存储器的端口位宽一致，通过原操作数对应字节之间进行相关操作，即

可实现BS×4个字节的相关运算。

[0080] 本实施例中主要介绍在LTE中MIMO预编码操作中的开环空间复用的具体操作流程，其中码本矩阵行列都等于4，通过对多个矩阵相乘操作的向量化处理来加快运算速率。

[0081] 由于4天线开环空间复用中$W(i)=C_k$，$k=\left(\left\lfloor\frac{i}{\upsilon}\right\rfloor \bmod 4\right)+1\in\{1,2,3,4\}$，其中$C_1$，$C_2$，$C_3$，$C_4$从码本索引为12、13、14、15中选择。D(i)为周期性变化的矩阵，U为一个固定矩阵。因此，$W'(i)=W(i)\times D(i)\times U$是一个在16个矩阵中按一定规律选取的矩阵。本发明提前将这16个矩阵算出，并按照该规律在多粒度并行存储器中排列。

[0082] 在本发明实施例中，码本矩阵和信号矩阵在多粒度并行存储器中排列规则如图2和图3所示。

[0083] 本发明实施例提供了一种在开环空间复用中大规模矩阵乘法的向量化实现方法。该方法包括：

[0084] 步骤2001：从多粒度并行存储器10中读取16行码本数据，存入寄存器A_1～A_{16}中，并读取4行信号数据，存入寄存器B_1～B_4中。

[0085] 步骤2002：利用向量处理部件执行$E=A_1\times B_1+A_2\times B_2+A_3\times B_3+A_4\times B_4$，然后将结果E存入多粒度并行存储器中，得到天线0上的16个数据；

[0086] 步骤2003：利用向量处理部件执行$E=A_5\times B_1+A_6\times B_2+A_7\times B_3+A_8\times B_4$，然后将结果E存入多粒度并行存储器中，得到天线1上的16个数据；

[0087] 步骤2004：利用向量处理部件执行$E=A_9\times B_1+A_{10}\times B_2+A_{11}\times B_3+A_{12}\times B_4$，然后将结果E存入多粒度并行存储器中，得到天线2上的16个数据；

[0088] 步骤2005：利用向量处理部件执行$E=A_{13}\times B_1+A_{14}\times B_2+A_{15}\times B_3+A_{16}\times B_4$，然后将结果E存入多粒度并行存储器中，得到天线3上的16个数据；

[0089] 步骤2006：从多粒度并行存储器中读取接下来的4行信号数据，存入寄存器B_1～B_4中，重复步骤2002～2005。直到所有信号数据全部运算完毕。

[0090] 以上所述的具体实施例，对本发明的目的、技术方案和有益效果进行了进一步详细说明，应理解的是，以上所述仅为本发明的具体实施例而已，并不用于限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

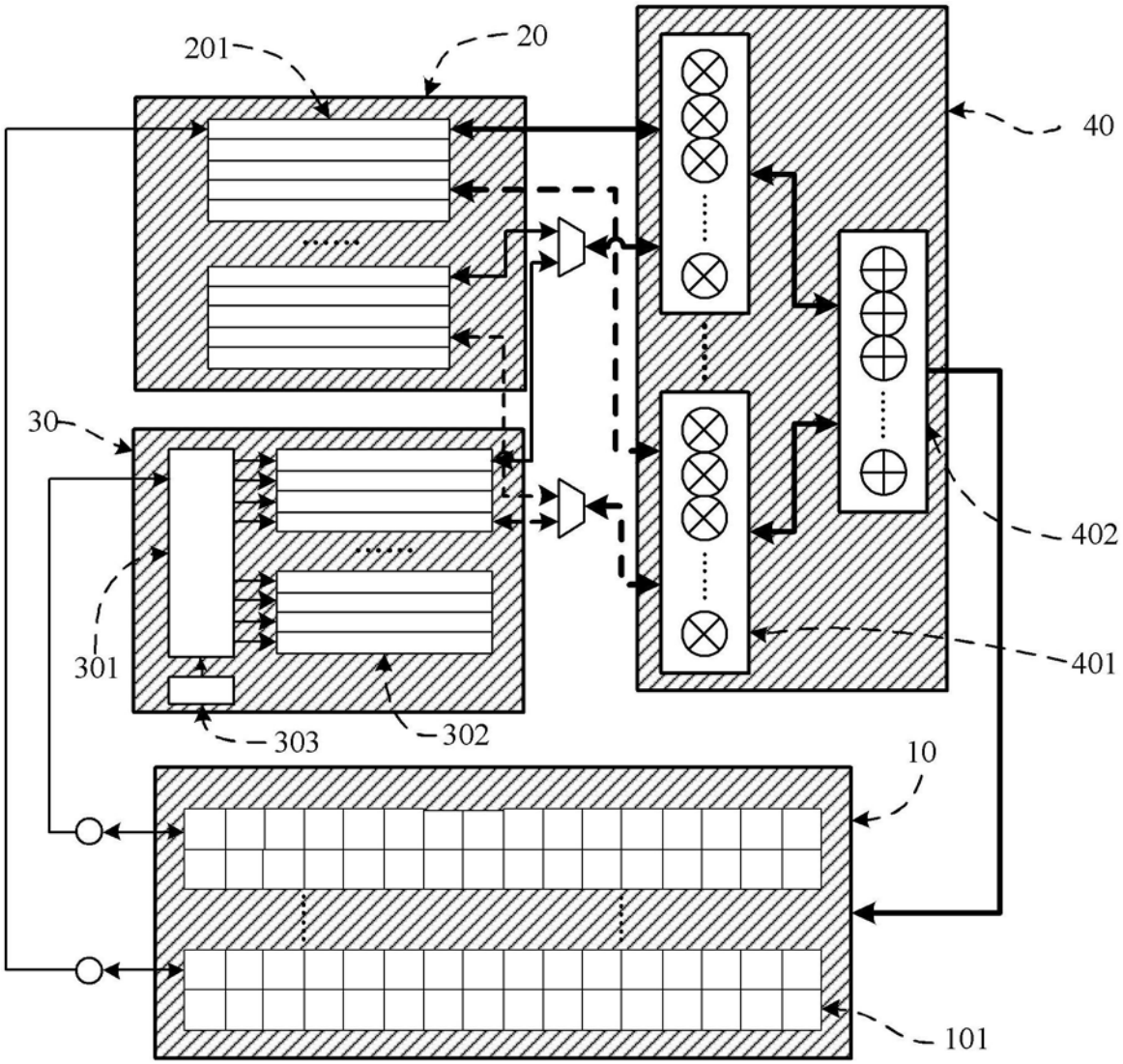


图1

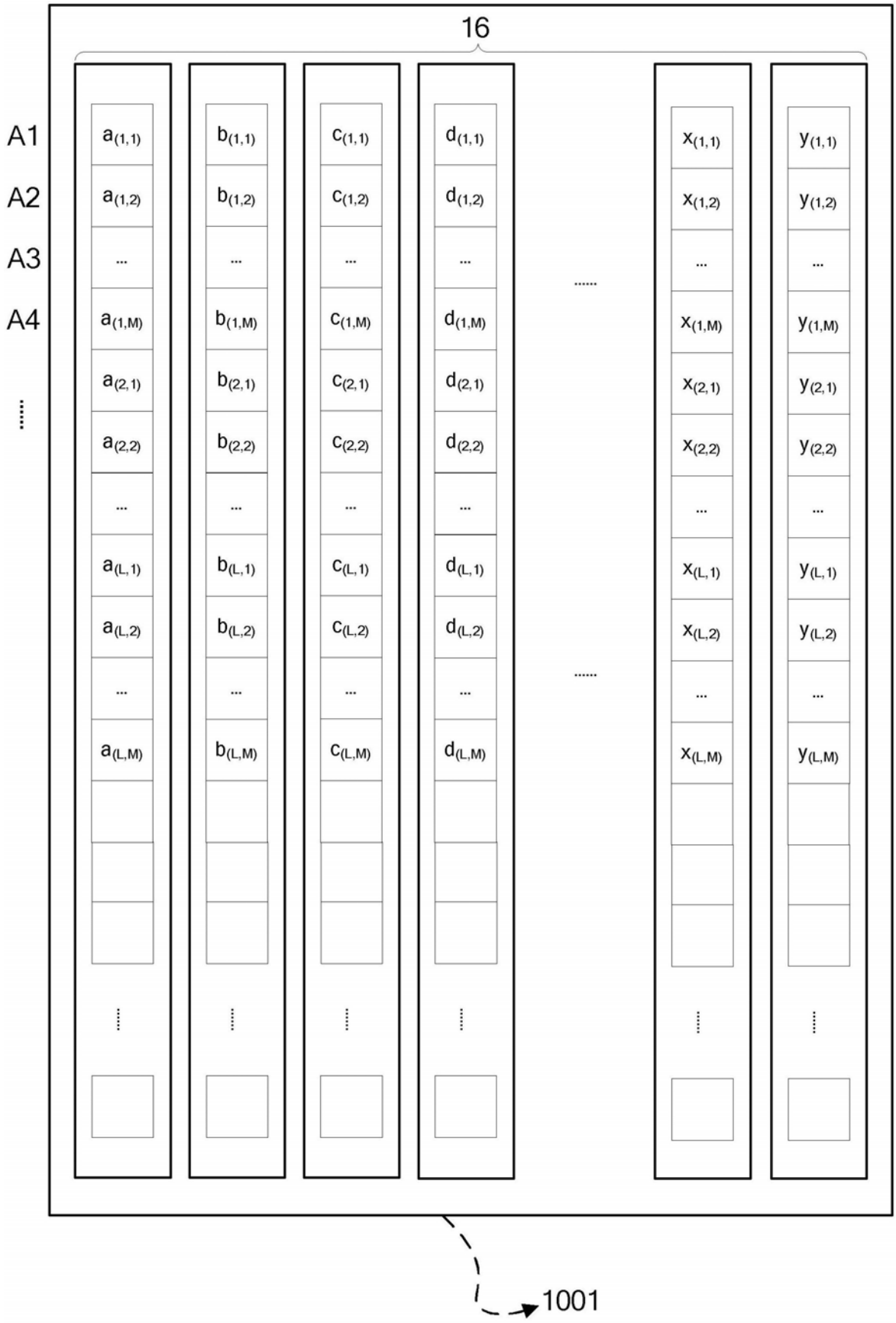


图2

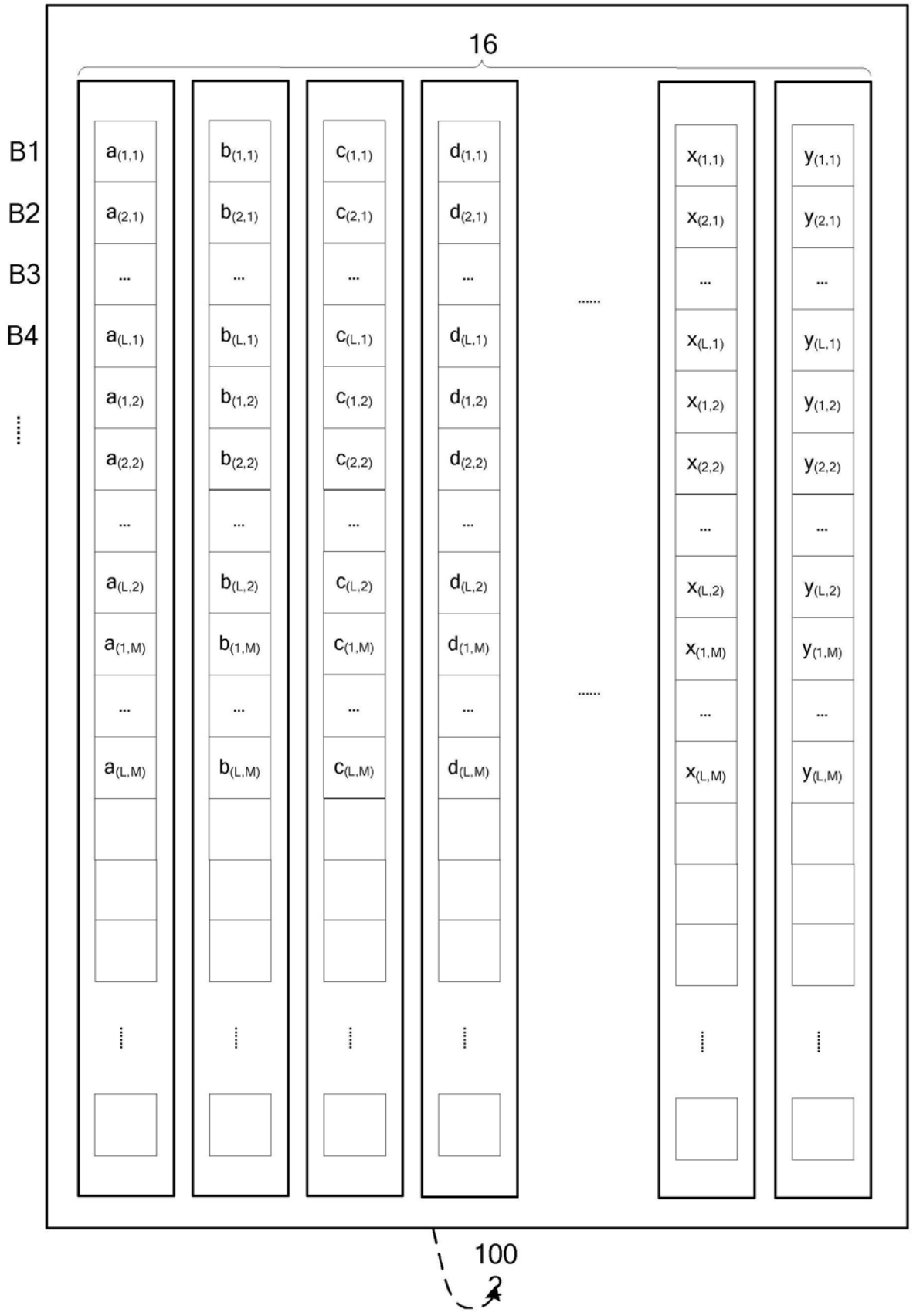


图3

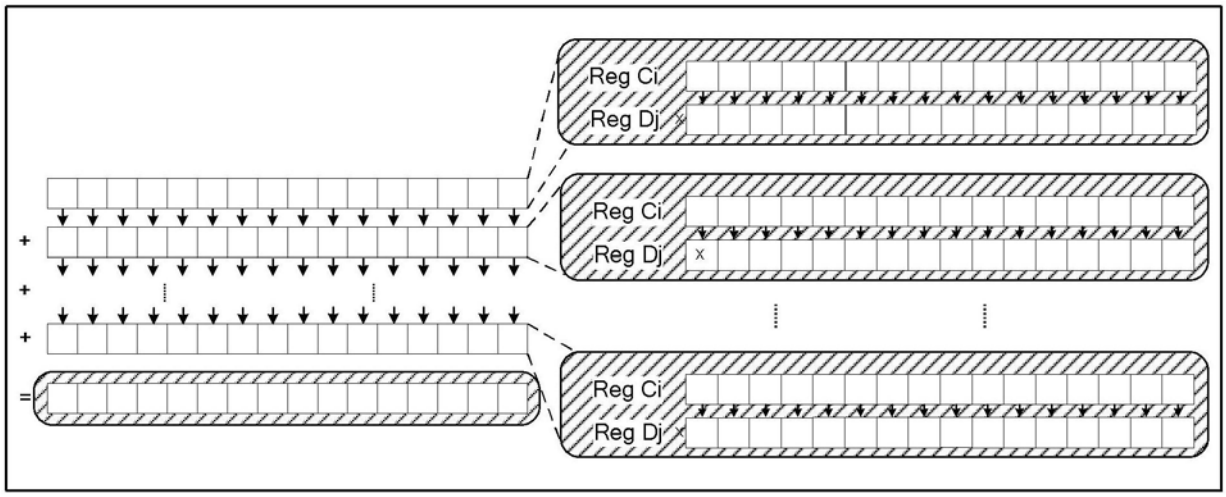

Reg Ci
Reg Dj
Reg Ci
Reg Dj
Reg Ci
Reg Dj
+
+
+
=

图4

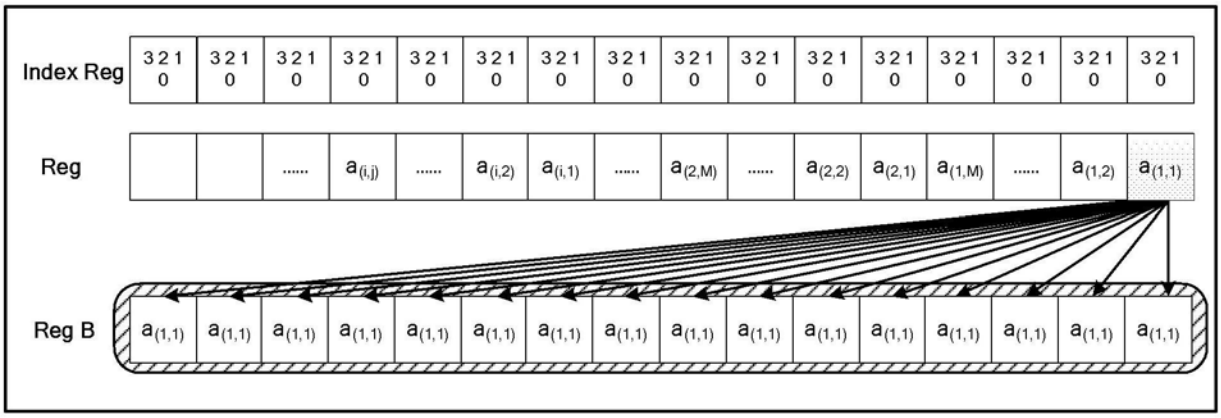

Index Reg
3 2 1 0
Reg
a(i,j)
a(i,2)
a(i,1)
a(2,M)
a(2,2)
a(2,1)
a(1,M)
a(1,2)
a(1,1)
Reg B
a(1,1)

图5