（由本局填寫）

| | |
|---|---|
| 承辦人代碼： | A6<br>B6 |
| 大　　類： | |
| ＩＰＣ分類： | |

本案已向：

| 國（地區） | 申請專利，申請日期： | 案號： | ，□有　□無主張優先權 |
|---|---|---|---|
| 美國 | 2000年10月06日 | 09/680,665 | ☑有□無主張優先權 |

有關微生物已寄存於：　　，寄存日期：　　，寄存號碼：

（請先閱讀背面之注意事項再填寫本頁各欄）

裝　訂　線

五、發明說明（ 1 ）

## 技術範疇

本發明係有關於數位信號處理的領域。它可提供用以執行線性轉換之一改良方法及裝置，且具減少算術運算次數、簡化電路、與低消耗功率。

## 發明背景

線性轉換普遍用於信號或資料處理。一線性轉換可從認為"輸入向量"的一n維矩陣向量、及認為"輸出向量"的一r維矩陣向量產生。在許多應用中，輸入向量是由需要處理的一特定信號數位取樣組成。然而，線性轉換應用並未受限於任何特殊領域，它們本質可用於科學與技術的每個領域，而且他們有效性能是高度想要。

從n維矩陣向量組(實數、複數、或任何欄位的數量)到r維矩陣向量組(相同領域的數量)的一般線性轉換可藉由下列排列r×n提供：

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{r1} & & & a_{m} \end{bmatrix}$$

一般輸入n維矩陣欄向量可寫成下列形式：

五、發明說明（ 2 ）

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

該線性轉換可藉由下列公式提供的矩陣乘積而從輸入向量x及r維矩陣的一輸出欄向量y產生：

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_r \end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_{r1} & & & & & & a_m \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}=$$

$$=\begin{bmatrix} a_{11}x_1+ & a_{12}x_2+ & \cdot & \cdot & \cdot & \cdot & +a_{1n}x_n \\ a_{21}x_1+ & a_{22}x_2+ & \cdot & \cdot & \cdot & \cdot & +a_{2n}x_n \\ \cdot & \cdot & & & & & \cdot \\ \cdot & \cdot & & & & & \cdot \\ \cdot & \cdot & & & & & \cdot \\ a_{r1}x_1 & a_{r2}x_2+ & \cdot & \cdot & \cdot & \cdot & +a_m x_n \end{bmatrix}$$

線性轉換的一簡單性能需要r・n乘積及r-(n-1)加法。

二進位矩陣在本發明的發展與應用方面是特別重要。二進位相反矩陣是在他們的登錄中只包含±1值。此後，他們稱為U矩陣。由一U矩陣表示的線性轉換稱為U轉換。在上述中，如果特定轉換是一U轉換，那麼r-(n-1)加法/減算於簡單性能是需要的。另一類型二進位矩陣在他們登錄中只包含0、1值。他們稱為0-1矩陣。由0-1矩陣所表示的一線

五、發明說明（ 3 ）

性轉換稱為一0-1轉換。在上述中，一般而言，r-(n-1)/2加法於一0-1轉換的簡單性能是需要。

在本文中，術語"二進位轉換"是由上述兩類型轉換所組成：U轉換及0-1轉換。為了要完成引用提及的技術，術語U向量可視為在它的元件中具有±1值，同樣地，在它元件中具有0、1值的一向量將可稱為一0-1向量。

二進位轉換處理是由輸入向量的加法與減算元件所組成。他們能以需要例如加法器及減法器的元件之一電子應用特殊積體電路(ASIC)硬體實施。這些元件的使用是昂貴且消耗能量，而且他們結構會佔用一大區域。具一低資源消耗的二進位轉換硬體實施在許多技術領域是逐漸增加需要。

在數位信號處理的各種不同觀點中有廣泛使用的U轉換。它包括例如影像處理與無線通訊的多種通訊技術。

現階段，全世界持續熱衷於直接序列(DS)劃碼多工存取(CDMA)展佈頻譜通訊。IS-95 [TIA/EIA/IS-95-A, "Mobile Station Base Station Compatibility Standard for Dual-Mode Wideband Spread spectrum Cellular System, Feb 27, 1996]是發展DS-CDMA系統的一範例。

在CDMA傳輸技術中，一多使用者資料可在一合成信號中傳送，然後在傳輸之前乘以一假隨機雜訊(PN)碼，其是一具有隨機雜訊性質(例如低相關係)之一U序列。展佈性質可產生抗雜訊的傳輸，包括多路徑雜訊、干擾或偵測。長於信道碼的一捲積碼亦可使用。第二代(IS-95-B)與第三代(3G)寬頻帶(WB)CDMA標準的傳輸方式需要接收器執行多

五、發明說明（ 4 ）

碼偵測。此是同時將組合頻道解除展佈的一工作，其每個頻道是先前展佈成一不同信道碼。此可藉由U轉換應用達成，其中包含矩陣的展佈碼時常是一Hadamard矩陣的列。此許多範例之中的一者，其中有效U轉換技術於達成一低資源消耗的計算工作是想要的。

有數種已知機構可改善線性轉換的特殊性能。從一非常大範圍選取的一些相關範例將在此提及。執行捲積及於DSP可當作濾波器使用的Toplitz轉換可藉由使用傳統快速傅利葉轉換(FFT)演算法而有效達成，該演算法需要$O(n\cdot\log_2(n))$加法與乘法運算，其中n是領域空間的大小。對照下，該標準簡單方法需要$O(n^2)$加法與乘法。有關更詳細描述可參考：[James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Mathematics of computation, 19(90): 297-301, April 1965.]

在包括CDMA技術的數位信號處理中所使用的一特殊類型U轉換是由一Hadamard矩陣表示的Walsh-Hadamard轉換。一Hadamard矩陣可依下列循環定義：

$$H_1=[1],\quad H_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},$$

而且對於每個整數n而言，一2的乘冪：$H_{2n}=\begin{bmatrix}H_n & H_n\\ H_n & -H_n\end{bmatrix}$

用以執行的一低複數與能量保存演算法快速Walsh-Hadamard變換(FWT)，其可用以將n x n Hadamard矩陣的加

五、發明說明（ 5 ）

法/減法次數減到$n \cdot \log_2(n)$。此可在這些運算中提供一選擇性節省。在此方法下的基本觀念是類似FFT。

U轉換的觀點可藉由本發明的另一特性修改，以適於toplitz U轉換性能的有效方法及裝置。這些轉換係表示一特定U序列或一複數U序列的全部或部分捲積。Toplitz U轉換的無線通訊應用包括開始時間同步及使用具一實數或複數假隨機(PN)或黃金U序列的搜尋器。一黃金序列是由兩PN序列的$Z_2$加算組成。

本發明的上述二進位觀點可由本發明的另一觀點歸納出由一相當小數量不同登錄的rxn矩陣所表示一線性轉換性能之有效方法及裝置。本發明的進階觀點的應用包括由{0，1，-1}登錄的一矩陣所表示複數U轉換及線性轉換。本發明的廣泛較佳具體實施稱為一般化去除法(GEM)。

寬頻帶CDMA及進階DSP未來技術的可能其他分枝將可從不同展開因素的一輸入信號同時分析而獲得。此可將方法提供一多碼頻道，以便允許同時接待例如傳真、正常電話交談及網際網路資料的不同類型資訊，而不致於使網路過載。本發明的另一觀點是這些類型列型運算的有效方法及裝置。它包含具額外處理器的本發明U二進位觀點的修改版本。此額外處理器係使用有關該U二進位方法所需的加法次數，以發覺一結構而可在整個低速率加法中各種不同部分中運用U二進位方法。

本發明的額外主要應用包括無列多媒體系統個人衛星行動系統、GPS衛星定位系統等。在行動通訊及其他行動的

五、發明說明（ 6 ）

本文中，在用於線性處理以減少電流消耗的計算系統是必要的。在行動電話技術中的本發明應用可延長電池壽命、減少電路、及縮短響應時間。

圖式之簡單說明

本發明的這些及其他特性與優點可從下列詳細描述及附圖而變得更顯然。

圖1其係根據本發明的一較佳具體實施例而描述更新由一裝置所採用一記憶體內容的操作圖，其中該裝置可使用0-1二進位轉換矩陣執行轉換；

圖2係根據本發明的一較佳具體實施例而描述更新由一裝置所採用一記憶體內容的操作圖，其中該裝置可使用U轉換矩陣執行轉換；

圖3其係根據本發明的一較佳具體實施例而描述更新由一裝置所採用一記憶體內容的操作圖，其中該裝置可使用Toplitz轉換矩陣執行轉換；及

圖4係根據本發明的一較佳具體實施例的一裝置方塊圖，其中該裝置可用以執行減少加法次數的一線性轉換。

圖5係根據本發明的一具體實施例而描述一r×n U矩陣A與一n維矩陣向量X乘積的一實施範例。

圖式之詳細說明

有兩新術語是描述本發明方法的基本原理。第一是等值向量。如果他們之中每一是另一者的乘積，相同維矩陣的兩非零向量在隨後的本文中稱為同等。此術語可運用於一特定矩陣的兩列或兩欄。第二術語是在一特定矩陣中的一

五、發明說明（ 7 ）

欄先導元件。在本發明使用的定義是在加法矩陣各欄中的最高索引的非零元件。此定義是取決於在特定矩陣中一致性的一預先定義的欄索引。新順序的一重新定義可產生新先導元件的一重新定義。例如，一特定矩陣的先導元件可以是各欄的最底部非零元件或頂端非零元件，其是因類似於此目的所選取的順序而定。

0-1矩陣

本發明的較佳具體實施例可提供用以執行一0-1二進位線性轉換之有效方法。此後，它亦可稱為"0-1方法"。下列範例是描述0-1方法及其主要觀點的概述。

範例：提供一0-1二進位4×14矩陣A：

$$A=\begin{bmatrix}0&1&1&0&1&1&0&0&1&0&0&0&1&1\\1&0&0&1&0&1&1&0&0&0&1&1&1&0\\0&0&1&1&1&1&0&0&0&0&1&0&1&1\\1&1&1&1&0&1&1&0&1&0&1&1&1&1\end{bmatrix},$$

及一14維矩陣輸入向量，

$$x=\begin{bmatrix}x_1\\x_2\\\cdot\\\cdot\\\cdot\\x_{14}\end{bmatrix}$$

假設，它想要計算4維矩陣"輸出"向量：

五、發明說明（ 8 ）

$$y=\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=\begin{bmatrix}0&1&1&0&1&1&0&0&1&0&0&0&1&1\\1&0&0&1&0&1&1&0&0&0&1&1&1&0\\0&0&1&1&1&1&0&0&0&0&1&0&1&1\\1&1&1&1&0&1&1&0&1&0&1&1&1&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\cdot\\\cdot\\\cdot\\x_{14}\end{bmatrix}=A\bullet x.$$

根據本發明的一較佳具體實施例，矩陣A是先檢查相等列。既然沒有兩列是相等，所以程序會進行下一步驟。然後，輸出向量y能以欄與係數相加的表示，其中該等係數是相對輸入向量座標。因此：

$$y=x_1\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+x_2\begin{bmatrix}1\\0\\0\\1\end{bmatrix}+x_3\begin{bmatrix}1\\0\\1\\1\end{bmatrix}+x_4\begin{bmatrix}0\\1\\1\\1\end{bmatrix}+x_5\begin{bmatrix}1\\0\\1\\0\end{bmatrix}+x_6\begin{bmatrix}1\\1\\1\\1\end{bmatrix}+x_7\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+x_8\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+x_9\begin{bmatrix}1\\0\\0\\1\end{bmatrix}+$$
$$x_{10}\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+x_{11}\begin{bmatrix}0\\1\\1\\1\end{bmatrix}+x_{12}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+x_{13}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}+x_{14}\begin{bmatrix}1\\0\\1\\1\end{bmatrix}$$

在此階段，連同他們係數的零欄將可被刪除。經由與輸入向量有關的本發明之一較佳具體實施例達成的第一步驟可將係數歸納及加算。如此，隨著6個加算，下列換算的向量方程式便可完成：

五、發明說明（ 9 ）

$$y=(x_1+x_7+x_{12})\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+(x_2+x_9)\begin{bmatrix}1\\0\\0\\1\end{bmatrix}+(x_3+x_{14})\begin{bmatrix}1\\0\\1\\1\end{bmatrix}+(x_4+x_{11})\begin{bmatrix}0\\1\\1\\1\end{bmatrix}+x_5\begin{bmatrix}1\\0\\1\\0\end{bmatrix}+$$

$$(x_6+x_{13})\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

此表示可藉由下列定義而簡化：

$w_1=x_1+x_7+x_{12}$、$w_2=x_2+x_9$、$w_3=x_3+x_{14}$、$w_4=x_4+x_{11}$、$w_5=x_5$、$w_6=x_6+x_{13}$，其表示：

$$y=\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=w_1\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+w_2\begin{bmatrix}1\\0\\0\\1\end{bmatrix}+w_3\begin{bmatrix}1\\0\\1\\1\end{bmatrix}+w_4\begin{bmatrix}0\\1\\1\\1\end{bmatrix}+w_5\begin{bmatrix}1\\0\\1\\0\end{bmatrix}+w_6\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

此同等於最初問題，但是可減少欄數量。然而，現在此減少類型可交換。為了要獲得進一步增益，向量方程式可分成下列同等組的兩向量方程式：

$$\begin{bmatrix}y_1\\y_2\end{bmatrix}=w_1\begin{bmatrix}0\\1\end{bmatrix}+w_2\begin{bmatrix}1\\0\end{bmatrix}+w_3\begin{bmatrix}1\\0\end{bmatrix}+w_4\begin{bmatrix}0\\1\end{bmatrix}+w_5\begin{bmatrix}1\\0\end{bmatrix}+w_6\begin{bmatrix}1\\1\end{bmatrix}$$

$$\begin{bmatrix}y_3\\y_4\end{bmatrix}=w_1\begin{bmatrix}0\\1\end{bmatrix}+w_2\begin{bmatrix}0\\1\end{bmatrix}+w_3\begin{bmatrix}1\\1\end{bmatrix}+w_4\begin{bmatrix}1\\1\end{bmatrix}+w_5\begin{bmatrix}1\\0\end{bmatrix}+w_6\begin{bmatrix}1\\1\end{bmatrix}$$

這兩方程式之中每一者將可如此較佳具體實施例的第一步驟個別處理。如此，相同非零欄的係數將可歸納及加算。因此，在此階段上，該方法達成具6個加算二的兩向量方程式組：

五、發明說明（ 10 ）

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (w_1 + w_4)\begin{bmatrix} 1 \\ 0 \end{bmatrix} + (w_2 + w_3 + w_5)\begin{bmatrix} 1 \\ 0 \end{bmatrix} + w_6\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = (w_1 + w_2)\begin{bmatrix} 0 \\ 1 \end{bmatrix} + (w_3 + w_4 + w_6)\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

現在，很顯然只有4個額外加法需要完成輸出向量y的計算。因此，想要的結果可使用整個16個加算完成。執行此計算處理的傳統先前技藝方法需要28個加算。當矩陣的維矩陣成長時，本發明較佳具體實施例的前述處理將可顯示一成長相當有效方法。

大體上，本發明的0-1二進位於一n維矩陣向量x的r×n維矩陣的二進位0-1矩陣A乘積是一有效方法及裝置。該矩陣如下示：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{r1} & & & a_m \end{bmatrix}$$

其中 $a_{ij}=0$，1而 $1 \le i \le r$，$1 \le j \le n$。

而且該輸入向量如下示：

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

輸入向量的登錄可以是實數或複數或屬於任何欄位數量

五、發明說明（ 11 ）

(例如$Z_2$)。該目標是要藉由向量x而計算矩陣A乘積的結果。此結果將可由下式表示；

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_r \end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_{r1} & & & & & & a_m \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

將描述的程序是可重覆，而且下列一連串步驟或其一部分可於每個反複重複。根據本發明的一較佳具體實施例，第一步驟是檢查相等列。當可能時，此步驟應該在任何輸入向量處理開始之前完成，當作一預備。如果發現i列等於j列，那麼它可推測yi是等於yj。因此，兩相等列之中一者將可省略。為了清楚緣故，具有較大索引的列將可始終被省略。因此，如果j > i，那麼j列將可被省略。此初始運算可在本發明執行最後步驟上修改，其中在此階段(既然它是等於$y_i$)已知的$y_j$可回到向量y的適當位置。相等列的省略將可持續，直到他們在矩陣A沒有兩列相等為止。實際上，在階段在許多場合可跳過。每當它執行時，於相等列有一合理的可能性。既然此情況可確定相等列的存在，所以當$\log_2(r)>n$時，它始終可被執行。

為了要避免麻煩的符號表示，當r×n維矩陣的最初矩陣與名稱亦要維持時，從此隔離處理產生的修改矩陣將具有相同名稱A。在下一階段，一加總處理可用來移除相等欄。此程序可在最少加算的修改矩陣中減少欄數量。首先，向

五、發明說明（ 12 ）

量乘積y=A•x的矩陣可分解成A欄乘以相對x元件的加算。因此：

$$y = A \bullet x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ . \\ . \\ . \\ a_{r1} \end{bmatrix} + x_2 \begin{bmatrix} a_{11} \\ a_{21} \\ . \\ . \\ . \\ a_{r2} \end{bmatrix} + \ldots\ldots\ldots\ldots\ldots\ldots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ . \\ . \\ . \\ a_m \end{bmatrix}$$

此加算的每個元件是由矩陣A的一欄向量所組成，其中該矩陣A是乘以向量x的對應數量係數。此加算可簡潔寫成：$y=A\bullet x=x_1v_1+x_2v_2+\ldots+x_nv_n$

其中(每j)$v_j$是A的j欄：$v_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ . \\ . \\ . \\ a_{rj} \end{bmatrix}$

根據本發明的一較佳具體實施例，零欄及其係數可在此階段省略。其次，相同非零欄可組匯集及重配置，其中每個獨特欄可變成一共同因素，其中該共同因素是乘以對應數量係數的加總。因此，結果加總可藉著重複欄的省略而包含從最初一連串非零欄擷取一連串的所有非零欄$w_1,\ldots\ldots,w_m$。各欄wj是乘以一係數$t_j$，其中該係數$t_j$是等於此欄的所有欄的對應最初係數之一加總。很清楚，$m\leq n$，而且差異n-m是最初序列$v_1,v_2,\ldots\ldots,v_n$中反複數目。上述處理可由下列數學描述而形成公式：

五、發明說明（ 13 ）

$$y = A \bullet x = \square_{1 \le j \le n} x_j v_j$$

$$= \square_{1 \le k \le m} \square_{j \text{ such that: } v_j = w_k} x_j v_j$$

$$= \square_{1 \le k \le m} \left( \square_{j \text{ such that: } v_j = w_k} x_j \right) w_k$$

目前，對於所有1≤k≤m而言，定義：$t_k = \square_{j \text{ such that: } v_j = w_k} x_j$，以致於：

到現在為止，計算工作是此新係數$t_1,.....,t_m$的計算，如同最初一些$x_1,.....,x_n$的加總。n-m加算是需要。乘積y=A•x 如此可依下列提供：

$$y = A \bullet x = \square_{1 \le k \le m} t_k w_k$$

用以決定需要加總元件的計算管理觀點可於輸入向量出現之前可於每個矩陣一次達成，當作一預備。概括而言，讓B是r×m矩陣，其中該矩陣的欄分別是$w_1,.....,w_m$，而且讓：

$$t = \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ t_m \end{bmatrix}$$

然後y=A•x=B•t，如此，上述處理可將r×n矩陣A乘以n向量x的最初問題可減少到r×m矩陣B乘以m向量t的問題。當m較小時，此允許的增益會較大。

下一步驟是將先從階段的矩陣B水平分成兩或多個子矩陣，其中列是保持完整。每個子矩陣將具有減少的列數量，而且可於下一重複中提高上述欄歸納程序的增益。矩陣B

五、發明說明（ 14 ）

的列可由下列表示：$u_1$，$u_2$，.....，$u_r$。考慮下的乘積然後可依下列表示：

$$y = B \bullet t = \begin{bmatrix} u_1 \cdot t \\ u_2 \cdot t \\ . \\ . \\ . \\ u_r \cdot t \end{bmatrix}$$

$u_1 \cdot t$

其中，每列是在數量乘積中乘以向量t。鑑於此表示，計算y=B•t的工作是同等於計算r數量乘積的組合工作：$u_1$•t, $u_2$•t,.....,$u_r$•t。

根據本發明的一較佳具體實施例，此可藉由計算向量乘積的矩陣而達成：$B_1$•t和$B_2$•t，其中矩陣$B_1$和$B_2$之中每一者包含矩陣B的一部分列，而且這兩部分是互相散開，而且他們的聯集包含B中所有列組。通常，除了第一重複之外，兩子矩陣將具有相同或幾乎相同數量的列。然而，在一些特殊情況，可分成超過兩矩陣，其是因矩陣A的性質而定。此情況是當矩陣A沒有特殊內部順序時，它的登錄可認為是隨機選取。然後，本發明的一較佳具體實施例將暗示第一列分割將造成所有分開的子矩陣$\log_2$(欄數量）>列數量的狀態。然後，處理反複可將列分成量幾乎相等兩部分。矩陣認為隨機的情況本質是最壞情況。

可取代上述水平分割插入或在其前的另一步驟是一垂直分割。根據本發明的一較佳具體實施例，上述加總如下所

五、發明說明（ 15 ）

示：

$$y=\square_{1\le k\le m} t_k w_k$$

可分成兩部分：

$$y=\square_{1\le k\le p} t_k w_k + \square_{p+1\le k\le m} t_k w_k$$

其中p是在1與m-1之間選取。如此，對於$B_1$而言，矩陣的欄組是$w_1,.....,w_p$，而且對於$B_2$而言，矩陣的欄組是$w_{p+1},.....,w_m$，它可保持：

$$y=B_1 \cdot t' + B_2 \cdot t''$$

其中，對應向量是：$t'=(t_1,.....,t_p)$和$t''=(t_{p+1},.....,t_m)$。因此，藉由本發明的較佳具體實施例，兩較低維矩陣乘積$B_1 \cdot t'$和$B_2 \cdot t''$將可個別計算，而且最後，兩r向量的結果可一起加總。B欄$w_1,.....,w_m$索引的一初步再配置可提高此步驟的效力。

垂直分割將比其他程序較少使用。它的主要使用是在列數量實質超過欄數量的情況，較少使用於DSP應用情況。此步驟的基本缺點是需要將兩乘積$B_1 \cdot t'$與$B_2 \cdot t''$的結果加總，其中該等乘積是由在上述水平分割中沒有平行的一額外組r數量加算所組成。類似上述水平列分割，垂直分割可分成超過兩矩陣，其是因矩陣性質而定。

最後，該等上述步驟之中每一者可應用在重複的反複，如此可形成一循環機構。

然後，一範圍可提供上述方法所允許的節省。對於一r×n

五、發明說明（ 16 ）

0-1矩陣A，以致於$r<\log_2(n)$而言，在最壞情況由s*(n, r)表示的加算次數可以下式表示：

$$s^*(n, r) = n + s^*(r).$$

下表可將範圍提供給具許多列的矩陣加算次數。

| | |
|---|---|
| $s^*(2) = -1$ | $s^*(n, 2) = n - 1$ |
| $s^*(3) = 2$ | $s^*(n, 3) = n + 2$ |
| $s^*(4) = 7$ | $s^*(n, 4) = n + 13$ |
| $s^*(5) = 22$ | $s^*(n, 5) = n + 49$ |

下列是最壞情況的範圍：

$$s^*(r) < 2^r + 2^{r/2+2} - r.$$

標準(先前技藝)方法需要u(A)-r加算，其中u(A)是在矩陣A的數值1'ns。一般而言，(預期當A是隨機)$u(A)=(n-2)\cdot r/2$。既然當n是無窮大時，所以$s^*(n,r)/n$近似1，而且r是常數(或$r \le c\cdot\log_2(n)$，其中$1>c>0$)，此方法漸近是最理想(對於範圍r及無範圍n)。

當A是一r×n 0-1矩陣，且未假設矩陣A的r×n時，那麼本發明所需計算乘積A•x的加算次數可限制在$(1+\square)n\cdot r/\log_2(n)$，其中$1>\square>0$，而且當r和n是無窮大時，□會是零。在r>n的情況，一較小範圍(有關此情況)會存在，此情況是從垂直分割的應用產生：

$$(1+\square)n\cdot r/\log_2(r)，其中 1>\square>0，$$

而且當r和n是無窮大時，□會是零。

五、發明說明（ 17 ）

為了要評估此處理效率，可看出本發明可提高管理運算及減少加算。然而，加算是複雜度的一主要部分，而且經由使用本發明可減少加算效果。而且，大部分管理運算可在資料向量處理之前於每矩陣A次達成，該建議方法可實質節省功率消耗與電路。當特定矩陣具有一些稀疏程度時，上述0-1二進位觀點可特別有效。此時常會遇到當此具體實施例的應用係結合在本發明的實數矩陣觀點中所述的一般實數矩陣與一向量乘積計算的分配算術。

此外，本發明的觀點可適於代數碼(參考：[Shu Lin, Daniel J. Costello, Jr "Error Control Coding Fundamentals and Applications "Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983])，其中一$Z_2$矩陣的編碼及解碼是乘以一$Z_2$向量(或$Z_{2n}$向量)。此在任何情況是同樣的，其中一$Z_2$矩陣是乘以一$Z_{2n}$向量。最後，除了特性2的領域之外，上述0-1二進位具體實施例及下一U二進位具體實施例可在數量的每個領域中容易互換。在每個特殊情況，更有效具體實施例可選取。

U矩陣

本發明(此後亦稱為"U方法")的U二進位較佳具體實施例描述所需的初步觀念是向量類似觀念。如果一是另一者的乘積，相同維矩陣的兩非零向量可稱視為同等。在此具體實施例的本文中，注意，如果兩U向量是相同、或如果他們之中一者是另一者的(-1)乘積，他們是同等。下列較佳具體實施例是相當類似上述一者。主要不同在於下列本文移

五、發明說明（ 18 ）

除的完成是同等於(列或欄)向量，而不是相等向量。此可加速移除率，如此便可提高效率。此具體實施例是經由一範例說明，說明觀念的本值：

範例：考慮下列5×13 U矩陣A與一13維矩陣輸入向量x的乘積。

矩陣可藉由下式提供：

$$A=\begin{bmatrix} -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

而且輸入向量可由下式提供：

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{13} \end{bmatrix}$$

乘積的結果可藉由下式提供：

$$y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}=A\bullet x$$

處理的第一步驟是要檢查同等列。實際上，它是在輸入向量達成之前每次完成一操作。一掃描可發覺特定矩陣的

五、發明說明（ 19 ）

相等列發生，列2是與列4相等，事實上，列4是與(-1相同)乘積列2相等。它遵守$y_4=-y_2$。因此它不必需計算$y_4$和$y_2$，因此，$y_4$的計算將可省略。如此，列4可從矩陣A刪除。結果的矩陣A'可依下列提供：

$$A'=\begin{bmatrix}-1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1\\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1\\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1\\ -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1\end{bmatrix}$$

對應輸出可由下式提供：

$$y'=\begin{bmatrix}y_1\\ y_2\\ y_3\\ y_4\end{bmatrix}$$

而且它可保持：y'=A'•x

此是第一減少。在下面，乘積A'•x可分解成A'欄乘以x元件之一加總。因此：

$$\begin{bmatrix}y_1\\ y_2\\ y_3\\ y_5\end{bmatrix}=x_1\begin{bmatrix}-1\\ 1\\ 1\\ -1\end{bmatrix}+x_2\begin{bmatrix}1\\ -1\\ 1\\ 1\end{bmatrix}+x_3\begin{bmatrix}-1\\ -1\\ -1\\ -1\end{bmatrix}+x_4\begin{bmatrix}-1\\ 1\\ 1\\ -1\end{bmatrix}+x_5\begin{bmatrix}1\\ 1\\ -1\\ 1\end{bmatrix}+x_6\begin{bmatrix}1\\ -1\\ -1\\ 1\end{bmatrix}+x_7\begin{bmatrix}-1\\ 1\\ -1\\ -1\end{bmatrix}+$$
$$x_8\begin{bmatrix}1\\ -1\\ 1\\ 1\end{bmatrix}+x_9\begin{bmatrix}1\\ -1\\ -1\\ 1\end{bmatrix}+x_{10}\begin{bmatrix}-1\\ 1\\ -1\\ -1\end{bmatrix}+x_{11}\begin{bmatrix}1\\ -1\\ 1\\ 1\end{bmatrix}+x_{12}\begin{bmatrix}-1\\ -1\\ 1\\ 1\end{bmatrix}+x_{13}\begin{bmatrix}1\\ 1\\ 1\\ 1\end{bmatrix}$$

然後，上述加總的該等欄向量之中每一者的正常化可實施，以致於每個修改向量的上面元件是1。如此，如果向量

五、發明說明（ 20 ）

的上面元件是(-1)，向量及其係數可皆乘以(-1)。然而，如果向量的上面元件是1，便沒有修改。上述加總的下列正常化形式可從下列取得：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_5 \end{bmatrix} = (-x_1)\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + x_2\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + (-x_3)\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + (-x_4)\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + x_5\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} + x_6\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + (-$$

$$x_7)\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} +$$

$$x_8\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + x_9\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + (-x_{10})\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + x_{11}\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + (-x_{12})\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} + x_{13}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

在此正常化加總中，向量的上面元件是1，而且不同向量的數量可減少。下一步驟將可歸納及加總一些欄係數。如此，隨著8個加算，下列方程式便可獲得：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_5 \end{bmatrix} = ((-x_1) + (-x_4) + x_6 + x_9)\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + (x_2 + (-x_7) + x_8 + (-x_{10}) + x_{11})\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + ((-x_3) +$$

$$x_{13})\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + x_5\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} + (-x_{12})\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

新係數現將可定義如下：

五、發明說明（ 21 ）

$w_1=(-x_1)+(-x_4)+x_6+x_9$, $w_2=x_2+(-x_7)+x_8+(-x_{10})+x_{11}$,

$w_3=-x_3+x_{13}$, $w_4=x_5$. $w_5=-x_{12}$

因此，上述方程式能以下列形式提供：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_5 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

係數：$w_1$、$w_2$、$w_3$、$w_4$、$w_5$在此階段是處理器已知的。在下一階段中，此向量方程式將可水平分成下列一組兩方程式：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_5 \end{bmatrix} = w_1\begin{bmatrix} -1 \\ 1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_4\begin{bmatrix} -1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

現在，這些方程式之中每一者能以反複的相同方式個別處理。既然它繼承先前方程式的正常狀態，所以上面方程式不需要正常化，如此，正常化只需用於第二方程式。因此，結果的組如下示：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_5 \end{bmatrix} = (-w_1)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix} + (-w_4)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + (-w_5)\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

在下一步驟中，相同向量的係數可歸納及加總。如此，隨著額外6個加算，結果是如下示：

五、發明說明（ 22 ）

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (w_1 + w_2)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + (w_3 + w_4 + w_5)\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_5 \end{bmatrix} = (-w_1 - w_4)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + (w_2 + w_3 + -w_5)\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

很清楚，向量y'可使用4個加算找到。為了要獲得最初輸出向量y，現只要回想$y_4 = -y_2$。如此，整個18個加算於此處理是需要的。可看出，傳統先前技藝方法需要60個加算來計算輸出向量。

本發明的U二進位較佳具體實施例是一r×m U矩陣A與一n維矩陣輸入向量x乘積的一有效方法。它通常可提供大於0-1較佳具體實施例的增益，而且具有更多應用。

矩陣A可依下式提供：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{r1} & & & a_m \end{bmatrix}$$

其中$a_{ij} = \pm 1$而$1 \le i \le r$，$1 \le j \le n$。

而且輸入向量可依下式提供：

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

輸入向量的登錄可以是實數、或複數、或屬於大於2特性

五、發明說明（ 23 ）

的任何欄位數量。目標是要計算下列乘積的結果：

$$y=A\bullet x。$$

第一步驟是一特定矩陣乘以相等列的移除，一步驟可以是一部分的預備。如果發現i列是等於j列，那麼它可歸納$y_i$等於$\pm y_j$。因此，本發明的第一步驟的一較佳具體實施例可決定兩相等列之中的一者可省略。為了清楚緣故，策略會是具有較大索引的列將始終是省略一者。因此，如果假設$j>i$，那麼j列可省略。此最初操作在處理的最後步驟可相反，其中在此階段已知的$y_j$可放回到向量y的原始位置。此去除處理可持續，直到修改的矩陣沒有相等兩列為止。

每當有相等列的一實際可能性時，此階段便可執行。此時常是U方法的多乘積(本發明的另一觀點)應用的情況。既然此情況可確保相等列的存在，所以當$\log_2(r)\geq n$時，此階段可始終執行。然而，有其他情況可確保相等列的存在，而且應該考慮。例如，此可以是Hadamard矩陣的子矩陣情況。為了要避免不靈活的表示法，結果修改的矩陣將具有與最初矩陣相同名稱A，而且維矩陣$r\times n$的名稱亦可保留。

下一步驟是去除相等欄。此將可減少使用最少加算修改的水平維矩陣(亦即欄數量)。與此步驟有關的所有管理可如0-1觀點達成，每次每個矩陣通常可在資料向量出現之前預備。首先，乘積$y=A\bullet x$可表示成A欄乘以對應x元件之一加總。

五、發明說明（ 24 ）

讓$v_j$(有關每個j)是A的j欄：$v_j=\begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{nj} \end{bmatrix}$ 然後：

$$A \bullet x = x_1 v_1 + x_2 v_2 + \ldots\ldots + x_n v_n$$

然後，如果它不是+1，矩陣A的各欄向量的上面元件可被檢查，而且正常化成+1值。因此，當$a_{ij}$=+1，那麼沒有正常化是需要；然而，如果$a_{ij}$=-1，那麼正常化會發生，而且對應j欄向量是乘以-1、及其對應係數$x_j$。上述加總的一新表示法如此便可獲得，其中每個向量的上面登錄是等於+1。在此表示法中，相同欄是始終相等。

相同欄然後可如上述0-1具體實施例而一起匯集，而且可藉由採用當作一共同因素的各欄而重排列，其中該共同因素是乘以它對應數量係數的加總。因此，產生的加總可藉由隔離反復正常化欄而包含從一連串正常化欄向量擷取的一連串所有獨特正常化欄向量$w_1$、.....、$w_m$(沒有重複)。每獨特正常化欄$w_j$是乘以一係數$t_j$，其中該係數$t_j$是所有正常化欄的對應正常化係數之一加總，而且該加總是等於此欄。很清楚m≤n，而且差n-m是在下列最初一連串正常化欄的重複次數目：$a_{11}v_1$、$a_{12}v_2$、.....、$a_{1n}v_n$。

算術處理如下示：

$$A \bullet x = \square_{1 \le j \le n} x_j v_j$$

$$= \square_{1 \le k \le m} \square_{j \text{ such that } a_{1j} v_j = w_k} (x_j a_{1j}) a_{1j} v_j$$

$$= \square_{1 \le k \le m} \left( \square_{j \text{ such that } a_{1j} v_j = w_k} a_{1j} x_j \right) w_k$$

五、發明說明（ 25 ）

對於所有1≤k≤m而言，可定義成：$t_k = \square_{j \text{ such that: } v_j = w_k} a_{lj} x_j$

在本發明此部分中包括的主要計算工作是計算新係數$t_1,.....,t_m$，當作正常化最初一些的加總。額外加算是n-m。乘積A•x可依下列提供：

$$A \cdot x = \square_{1 \le k \le m} t_k w_k$$

藉由欄去除處理的增益是因最初矩陣A的結構而定，它在無線電通訊的編碼及解碼信號向量中普遍使用的一些矩陣中是重要的，例如在Hadamard或循環PN矩陣的一些類型子矩陣。它實質亦是當矩陣A的列數量r相較於列數量n是相當小的情況。特別是r≤log(n)，在此情況，$m-n \ge n-2^{r-1} > 0$。這些看出對於0-1矩陣亦是同樣的。

本發明的其餘較佳具體實施例，水平與垂直分割、重複是相同於它0-1相對物。

為了要看出使用本發明所允許的節省，首先注意加算是複數的主要部分。對於一rxn U矩陣A而言，以致於在最壞情況由s(n, r)所表示的加算次數$r \le \log_2(n)$可界由下列提供：

$$s(n, r) = n + s(r)$$

其中：

$s(1) = -1$

$s(2) = 0$

$s(3) = 3$

$s(4) = 8$

$s(5) = 19$

$s(n, 1) = n - 1$

$s(n, 2) = n$

$s(n, 3) = n + 3$

$s(n, 4) = n + 8$

$s(n, 5) = n + 19$

五、發明說明（ 26 ）

$s(6) = 38$ $s(n, 6) = n + 38$

$s(7) = 75$ $s(n, 7) = n + 75$

$s(8) = 144$ $s(n, 8) = n + 144$

$s(9) = 283$ $s(n, 9) = n + 283$

下列範圍始終是有效：

$$s(r) < 2^{r-1} + 2^{r/2+1} - r$$

先前技藝傳統方法需要(n-1)•r加算。既然當n是無窮大，而r是常數時，s(n, r)/n近似1，所以此方法是漸近最理想。稍後所述本發明的更複數多乘積具體實施例對於最壞加算而言需要一精確的範圍。此將可藉由下列迴歸公式取得：

對於偶數r而言： $s(r)=2^{r-1} + 2s(r/2)$

對於奇數r而言： $s(r)=2^{r-1} + s((r+1)/2)+ s((r-1)/2)$

當一是r×n U矩陣，而且矩陣A的維矩陣r×n沒有限制，那麼在最壞情況計算乘積A•x的本發明所需加算次數始終是在範圍：$(1+\square)n \cdot r/\log_2(n)$，其中$1>\square>0$，而且當r和n是無窮大時，□是零。在r>n情況，一較小範圍(有關此情況)會存在，而且是從垂直分割的應用造成：

$(1+\square)n \cdot r/\log_2(r)$，其中$1>\square>0$，

而且，當r和n是無窮大時，□會是零。

然而，如果在矩陣A有一特殊類型結構，那麼計算矩陣A•x的本發明較佳具體實施例所需的加算次數可明顯降低。它的最一般情況是超過目前本文範圍，但是一些範例將提及。然而，在A是n×n Hadamard或循環假隨機(PN)矩陣的情況，那麼只有$n \cdot \log_2(n)$加算及n數量的記憶體將需要，其

五、發明說明（ 27 ）

在這一點上是最理想的。此說明亦適於0-1矩陣。

工業應用範例

有數種技術將可從上述本發明的較佳具體實施例實施獲益。一此技術是包括數個程序的影像處理，該等程序包括一U矩陣與一向量乘積。在無列通訊CDMA與IS-95、及在更進階第三代寬頻帶CDMA中，有數個處理是採用一U矩陣與一向量的乘積。這些處理將可使用無能量消耗電路完成，而且有時亦可藉著本發明的使用而消耗時間。

在IS-95-B的多碼的本文中，來自Hadamard-64的多碼本文的8列包含一U矩陣乘以一向量，其中該向量的登錄是一信號取樣。另一應用是藉由解展佈器的相鄰偵測。在此，存在者由一些Hadamard列組成的一矩陣乘以一資料向量的乘積。另一應用是搜尋器。它可藉由他們相互數量乘積計算而搜尋序列的較高相關性。相關器序列可從U序列的假隨機序列擷取，而且他們經由一資料向量的數量乘積是需要計算。初始調查是一搜尋器的範例。

一般化去除方法

本發明上述二進位觀點的特徵是重複下列處理：零列及同等列(其表示一是另一者的數量乘積)可省略；零欄可連同與他們有關的輸入向量數量元件省略；相等欄可匯集及加計，而且在這兩特徵用完之後，矩陣可水平或垂直分割。根據本發明的一較佳具體實施例，此步驟的反覆序列可運用在任何欄位數量的矩陣與向量的任何乘積。本發明的較寬廣較佳具體實施例將可稱為一般化去除方法(GEM)。

五、發明說明（ 28 ）

相等欄的加總基本上是下列處理的一重複。讓$v_1$、$v_2$、.....、$v_n$是轉換矩陣A的欄，而且$x=(x_1、x_2、.....、x_n)$是輸入向量，而且目標是要計算：$A•x$。假設在索引的一適當重排列之後，它可保持，其中該等欄$v_{k+1}$、.....、$v_n$之中每一者(對於一些$2\leq k<n$而言)是k欄的數量乘積，即是於$k<j\leq n$：$v_j=z_j v_k$。然後，n維矩陣向量x可藉由減少k維矩陣向量而取代：

$$x'=(x_1,x_2,.....,x_{k-1},x'_k) \quad 其中\ x'_k= x_k+x_{k+1}•z_{k+1}+.....+x_n•z_n$$

而且$r\times n$矩陣可藉由減少rxk矩陣A'取代，其中該矩陣A'的欄是$v_1$、$v_2$、.....、$v_k$。它然後可保持：$A'•x'=A•x$。根據本發明的一較佳具體實施例，如果需要索引變化，此處理可重複，直到沒有兩欄是相等為止。的確，此處理只是在本發明的U矩陣具體實施例中發生正常化處理的一般化。實際上，所有上述相等欄減少可同時完成。當此階段完成時，它便可認為是整個處理的一第一重複。然後，矩陣可水平分割，而且在每個分割中，上面欄去除能以在上述具體實施例達成的一循環方式重複。

當由U具體實施例及下一範例證明時，去除相等欄的一有效方法是藉由先將最上面元件的每欄分割、及將對應係數乘以一些最上面元件。結果，在每個修改欄的最上面元件具有1。然而，有時在GEM的應用中，轉換矩陣一欄的最上面元件是零，其是不可能使此分開。在此情況，簡單解決是藉由它最上面非零元件而將每非零欄分開，因此可將對應係數乘以相同最上面非零元件。結果在每個修改欄的

五、發明說明（ 29 ）

最上面非零元件是1。如果且只有如果他們是相等，此具有在修改矩陣中相等兩欄的想要結果。

為了要說明本發明的較佳具體實施例提高效率，可考慮在一相當小有限組S中具有它登錄的一矩陣有。實際上，此組可能是一有限欄位、一欄位乘算群的有限子集、或在乘算下的一接近欄的一有限子集，而且非常普遍是一欄位的任何有限子集。它包括上述的二進位情況，其中S={0，1}、或S={1，-1}，及其他非常普遍情況，其中S={0，1，-1}、或S={1，-1，j，-j}、或S={0，1，-1，j，-j}。增益可使用S大小降低，而且一般規則可保持GEM可將加算次數減少$\log_s(n)$的一因素，其中n是特定轉換矩陣的欄數量。當可適用時，GEM亦可與稍後討論的複數與一般具體實施例有效比較。

為了要描述此方法工作方式，它將證明登錄是來自下列組的一矩陣：$U_1$=def={1，-1，j，-j}，此稱為$U_1$矩陣。此類型矩陣時常實際出現。然而，它必須再次強調GEM並未局限於任何特殊類型的矩陣。加算次數可於r×n U1矩陣與一向量的乘積而局限於$C=n+4^{r-1}+o(4^{r-1})$範圍。

範例：可考慮下列3×15 $U_1$矩陣A與一15維矩陣複數輸入向量的乘積。此可依下式提供：

$$A=\begin{bmatrix} 1 & -j & j & -1 & 1 & -j & -1 & j & -1 & j & -1 & 1 & -j & 1 & j \\ -1 & j & -1 & j & 1 & j & 1 & 1 & j & -j & -j & -j & -j & -1 & -j \\ j & 1 & -1 & j & j & j & 1 & -1 & 1 & -1 & -1 & -j & 1 & j & -1 \end{bmatrix}$$

而且輸入向量可依下式提供：

五、發明說明（ 30 ）

$$x=\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_{15} \end{bmatrix}$$

乘積的結果是：

$$y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}=A\bullet x$$

在此範例達成的所有加算是複數加算。第一步驟是檢查相等列。沒有相等列可找到。在下一步驟中，乘積A•x可分解成A欄乘以相對x元件之一加總。

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}=x_1\begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix}+x_2\begin{bmatrix} -j \\ j \\ 1 \end{bmatrix}+x_3\begin{bmatrix} j \\ -1 \\ -1 \end{bmatrix}+x_4\begin{bmatrix} -1 \\ j \\ j \end{bmatrix}+x_5\begin{bmatrix} 1 \\ 1 \\ j \end{bmatrix}+x_6\begin{bmatrix} -j \\ j \\ j \end{bmatrix}+x_7\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

+

$$x_8\begin{bmatrix} j \\ 1 \\ -1 \end{bmatrix}+x_9\begin{bmatrix} -1 \\ j \\ 1 \end{bmatrix}+x_{10}\begin{bmatrix} j \\ -j \\ -1 \end{bmatrix}+x_{11}\begin{bmatrix} -1 \\ -j \\ -1 \end{bmatrix}+x_{12}\begin{bmatrix} 1 \\ -j \\ -j \end{bmatrix}+x_{13}\begin{bmatrix} -j \\ -j \\ 1 \end{bmatrix}+x_{14}\begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix}+$$

$$x_{15}\begin{bmatrix} j \\ -j \\ -1 \end{bmatrix}$$

然後，在上述加總中的該等欄向量之中每一者的正常化會發生。它可藉著將每個向量乘以它上面元件的倒數，及每個係數乘以對應向量的(相同)上面元件而達成。因此，

五、發明說明（ 31 ）

每個修改向量的上面元件是1。如果向量的上面元件是1，便不需要修改。下列的上述加總正常化形式可取得。

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + (-j)\cdot x_2 \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + j\cdot x_3 \begin{bmatrix} 1 \\ j \\ j \end{bmatrix} + (-1)x_4 \begin{bmatrix} 1 \\ -j \\ -j \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 1 \\ j \end{bmatrix} + (-j)x_6 \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + (-1)x_7 \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} +$$

$$+ j\cdot x_8 \begin{bmatrix} 1 \\ -j \\ j \end{bmatrix} + (-1)x_9 \begin{bmatrix} 1 \\ -j \\ -1 \end{bmatrix} + j\cdot x_{10} \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + (-1)x_{11} \begin{bmatrix} 1 \\ j \\ 1 \end{bmatrix} + x_{12} \begin{bmatrix} 1 \\ -j \\ -j \end{bmatrix} + (-j)x_{13} \begin{bmatrix} 1 \\ 1 \\ j \end{bmatrix} + x_{14} \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + j\cdot x_{15} \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix}$$

在此正常化加總中，每個向量的上面元件是1，而且當(在一般設定)欄n數量是充份相當大於列r數量(需求是：$n>4^{r-1}$)時，不同向量的數量如此可實質減少。下一步驟是歸納及加總相同欄的係數。如此，隨著7個加算，下列換算便可獲得：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = (x_1 + (-j)\cdot x_2 + j\cdot x_{10} + x_{14} \div j\cdot x_{15}) \begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + j\cdot x_3 \begin{bmatrix} 1 \\ j \\ j \end{bmatrix} + ((-1)x_4 + x_{12}) \begin{bmatrix} 1 \\ -j \\ -j \end{bmatrix} + (x_5 + (-j)x_{13} \begin{bmatrix} 1 \\ 1 \\ j \end{bmatrix} + ((-j)x_6 + (-1)x_7 \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}) + j\cdot x_8 \begin{bmatrix} 1 \\ -j \\ j \end{bmatrix} + (-1)x_9 \begin{bmatrix} 1 \\ -j \\ -1 \end{bmatrix} + (-1)x_{11} \begin{bmatrix} 1 \\ j \\ 1 \end{bmatrix}$$

新係數現將定義：

五、發明說明（ 32 ）

$w_1=x_1+(-j)\cdot x_2+j\cdot x_{10}+x_{14}+j\cdot x_{15}$, $w_2=j\cdot x_3$

$w_3=(-1)x_4+x_{12}$, $w_4=x_5+(-j)x_{13}$, $w_5=(-j)x_6+(-1)x_7$,

$w_6=j\cdot x_8$, $w_7=(-1)x_9$, $w_8=(-1)x_{11}$

因此，上述方程式能以下式寫成：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ -1 \\ j \end{bmatrix} + w_2\begin{bmatrix} 1 \\ j \\ j \end{bmatrix} + w_3\begin{bmatrix} 1 \\ -j \\ -j \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \\ j \end{bmatrix} + w_5\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + w_6\begin{bmatrix} 1 \\ -j \\ j \end{bmatrix} + w_7\begin{bmatrix} 1 \\ -j \\ -1 \end{bmatrix} + w_8$$

$\begin{bmatrix} 1 \\ j \\ 1 \end{bmatrix}$係數：$w_1,.....,w_8$在此階段是處理器已知的。然後，此向量方程式可水平分成下列相等組的兩方程式：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_2\begin{bmatrix} 1 \\ j \end{bmatrix} + w_3\begin{bmatrix} 1 \\ -j \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_6\begin{bmatrix} 1 \\ -j \end{bmatrix} + w_7\begin{bmatrix} 1 \\ j \end{bmatrix} + w_8$$

$\begin{bmatrix} 1 \\ j \end{bmatrix}$

$[y_3]=w_1[j]+w_2[j]+w_3[-j]+w_4[j]+w_5[-1]+w_6[j]+w_7[-1]+w_8[1]$

大體上，兩者是向量方程式。然而，在目前小範例中，第二方程式可退化成一數量方程式。規則目前是這些方程式之中每一者能以與第一重複的相同方式個別處理。既然上面方程式是繼承先前方程式的正常狀態，所以它不需要正常化，如此，在任何非退化應用中，正常化只需用於第二向量方程式。結果，上面向量方程式的結果換算如下示：

五、發明說明（ 33 ）

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (w_1 + w_5)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + (w_2 + w_8)\begin{bmatrix} 1 \\ j \end{bmatrix} + (w_3 + w_6 + w_7)\begin{bmatrix} 1 \\ -j \end{bmatrix} + w_4\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

此步驟需要4個更多加算量。計算目前能以兩方程式的一簡單使用額外13個加算完成。如此，在此範例中，本發明的較佳具體實施例係使用整個24個加算。布魯特斯力量計算將需要42個加算。在一較大規模中，節省是遠超過實質。

TOPLITZ矩陣

為了要證明本發明的較佳具體實施例，一範例將提出。

範例：假設長度8的一序列如下示：u=(1,1,-1,-1,1,-1,1,-1)，而且它需要檢查資料向量的3個連續假設：$x=(x_1,x_2,\ldots\ldots\ldots\ldots\ldots x_{10})$。目的可以是搜尋最大相關性。下列3個加總需要計算。

$y_1=1\cdot x_1+1\cdot x_2+(-1)\cdot x_3+(-1)\cdot x_4+1\cdot x_5+(-1)\cdot x_6+1\cdot x_7+(-1)\cdot x_8$

$y_2=1\cdot x_2+1\cdot x_3+(-1)\cdot x_4+(-1)\cdot x_5+1\cdot x_6+(-1)\cdot x_7+1\cdot x_8+(-1)\cdot x_9$

$y_3=1\cdot x_3+1\cdot x_4+(-1)\cdot x_5+(-1)\cdot x_6+1\cdot x_7+(-1)\cdot x_8+1\cdot x_9+(-1)\cdot x_{10}$

此可藉由下列toplitz矩陣乘積表示：

五、發明說明（ 34 ）

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 0 & 0 \\ -1 & -1 & 1 & -1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{10} \end{bmatrix}$$

$$= x_1\begin{bmatrix}1\\0\\0\end{bmatrix} + x_2\begin{bmatrix}1\\1\\0\end{bmatrix} + x_3\begin{bmatrix}-1\\1\\1\end{bmatrix} + x_4\begin{bmatrix}-1\\-1\\1\end{bmatrix} + x_5\begin{bmatrix}1\\-1\\-1\end{bmatrix} + x_6\begin{bmatrix}-1\\1\\-1\end{bmatrix} + x_7\begin{bmatrix}1\\-1\\1\end{bmatrix} + x_8\begin{bmatrix}-1\\1\\-1\end{bmatrix} + x_9\begin{bmatrix}0\\-1\\1\end{bmatrix} + x_{10}\begin{bmatrix}0\\0\\-1\end{bmatrix}$$

下一步驟是要匯集向量補數，如此：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = (x_1\begin{bmatrix}1\\0\\0\end{bmatrix} + x_9\begin{bmatrix}0\\-1\\1\end{bmatrix})(x_2\begin{bmatrix}1\\1\\0\end{bmatrix} + x_{10}\begin{bmatrix}0\\0\\-1\end{bmatrix}) + x_3\begin{bmatrix}-1\\1\\1\end{bmatrix} + x_4\begin{bmatrix}-1\\-1\\1\end{bmatrix} + x_5\begin{bmatrix}1\\-1\\-1\end{bmatrix} + x_6\begin{bmatrix}-1\\1\\-1\end{bmatrix} + x_7\begin{bmatrix}1\\-1\\1\end{bmatrix} + x_8\begin{bmatrix}-1\\1\\-1\end{bmatrix}$$

此時，可考慮該等角括弧之中每一者的內部項，及利用補助定理如果x、y是數量，而且v、u是向量，那麼：

$xv+yu=\frac{1}{2}(x+y)(v+u)+\frac{1}{2}(x-y)(v-u)$因此：

五、發明說明（ 35 ）

$$x_1\begin{bmatrix}1\\0\\0\end{bmatrix}+x_9\begin{bmatrix}0\\-1\\1\end{bmatrix}=\frac{1}{2}(x_1+x_9)\left(\begin{bmatrix}1\\0\\0\end{bmatrix}+\begin{bmatrix}0\\-1\\1\end{bmatrix}\right)+\frac{1}{2}(x_1-x_9)\left(\begin{bmatrix}1\\0\\0\end{bmatrix}-\begin{bmatrix}0\\-1\\1\end{bmatrix}\right)$$

$$=\frac{1}{2}(x_1+x_9)\begin{bmatrix}1\\-1\\1\end{bmatrix}+\frac{1}{2}(x_1-x_9)\begin{bmatrix}1\\1\\-1\end{bmatrix}$$

同樣地：

$$x_2\begin{bmatrix}1\\1\\0\end{bmatrix}+x_{10}\begin{bmatrix}0\\0\\-1\end{bmatrix}=\frac{1}{2}(x_2+x_{10})\begin{bmatrix}1\\1\\-1\end{bmatrix}+\frac{1}{2}(x_2-x_{10})\begin{bmatrix}1\\1\\1\end{bmatrix}$$

因此，隨著4個加算，結果如下示：

$$\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}=\frac{1}{2}(x_1+x_9)\begin{bmatrix}1\\-1\\1\end{bmatrix}+\frac{1}{2}(x_1-x_9)\begin{bmatrix}1\\1\\-1\end{bmatrix}+\frac{1}{2}(x_2+x_{10})\begin{bmatrix}1\\1\\-1\end{bmatrix}+\frac{1}{2}(x_2-$$

$$x_{10})\begin{bmatrix}1\\1\\1\end{bmatrix}+x_3\begin{bmatrix}-1\\1\\1\end{bmatrix}+x_4\begin{bmatrix}-1\\-1\\1\end{bmatrix}+x_5\begin{bmatrix}1\\-1\\-1\end{bmatrix}+x_6\begin{bmatrix}-1\\1\\-1\end{bmatrix}+x_7\begin{bmatrix}1\\-1\\1\end{bmatrix}+x_8\begin{bmatrix}-1\\1\\-1\end{bmatrix}$$

但是，現在上述U二進位方法可應用。如此，下一步驟是可使各欄的上面元件等於1的一正常化。

$$\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}=\frac{1}{2}(x_1+x_9)\begin{bmatrix}1\\-1\\1\end{bmatrix}+\frac{1}{2}(x_1-x_9)\begin{bmatrix}1\\1\\-1\end{bmatrix}+\frac{1}{2}(x_2+x_{10})\begin{bmatrix}1\\1\\-1\end{bmatrix}+\frac{1}{2}(x_2-$$

$$x_{10})\begin{bmatrix}1\\1\\1\end{bmatrix}+(-x_3)\begin{bmatrix}1\\-1\\-1\end{bmatrix}+(-x_4)\begin{bmatrix}1\\1\\-1\end{bmatrix}+x_5\begin{bmatrix}1\\-1\\-1\end{bmatrix}+(-x_6)\begin{bmatrix}1\\-1\\1\end{bmatrix}+x_7\begin{bmatrix}1\\-1\\1\end{bmatrix}+(-x_8)\begin{bmatrix}1\\-1\\1\end{bmatrix}$$

五、發明說明（ 36 ）

下一步驟是歸納及加總共同欄的係數。如此，隨著6個額外加算，我們可達成下列等式：

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = (\frac{1}{2}(x_1 + x_9) + (-x_6) + x_7 + (-x_8)) \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + (\frac{1}{2}(x_1 - x_9) + \frac{1}{2}(x_2 + x_{10}) + (-x_4)) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

$$+ \frac{1}{2}(x_2 - x_{10}) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + ((-x_3) + x_5) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

因此，如果我們為兩方便而定義下示：

$$w_1 = \frac{1}{2}(x_1 + x_9) + (-x_6) + x_7 + (-x_8), \qquad w_2 = \frac{1}{2}(x_1 - x_9) + \frac{1}{2}(x_2 + x_{10}) + (-x_4)$$

$$w_3 = \frac{1}{2}(x_2 - x_{10}), \qquad w_4 = \frac{1}{2}(-x_3) + x_5$$

然後，相等可如下寫成，

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = w_1 \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + w_2 \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + w_3 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + w_4 \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

在下一階段中，此向量方程式可水平分成下列組的兩方程式：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = w_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + w_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_3 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_4 \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$y_3 = w_1 + (-w_2) + w_3 + (-w_4)$$

藉由在第一方程式中歸納相同欄，我們可獲得兩額外加算：

五、發明說明（ 37 ）

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (w_1 + w_4)\begin{bmatrix} 1 \\ -1 \end{bmatrix} + (w_2 + w_3)\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

想要輸出向量y目前可使用5個額外加算找到。如此，具體實施本發明的方法需要17個額外加算以執行此計算。一布魯特斯力量方法將需要21個加算。很清楚，此小規模範例獲得的增益將不明顯，但是當我們看出在大規模應用的增益是與上述"平方"(非toplitz)U方法的增益是相當。

其次，本發明此觀點的一般設定描述。假設一U序列是依下列提供：

$$u_1, u_2, \ldots\ldots\ldots\ldots\ldots, u_n$$

而且，實數或複數數量的資料序列如下示：

$$x_1, x_2, \ldots\ldots\ldots\ldots\ldots, x_{n+m-1}$$

資料序列的元件可以是實數或複數或屬於大於2特徵的任何欄位特性數量。假設，它想要計算下列加總：

$$y_1 = u_1 \cdot x_1 + u_2 \cdot x_2 + \ldots\ldots\ldots\ldots\ldots + u_n \cdot x_n$$

$$y_2 = u_1 \cdot x_2 + u_2 \cdot x_3 + \ldots\ldots\ldots\ldots\ldots + u_n \cdot x_{n+1}$$

$$y_m = u_1 \cdot x_m + u_2 \cdot x_{m+1} + \ldots\ldots\ldots\ldots\ldots + u_n \cdot x_{n+m-1}$$

如果$m<\log_2(n)$，那麼讓$r=m$，否則，如果$m\geq\log_2(n)$，那麼本發明的一較佳具體實施例將可藉著將m個加總組分成r個連續加總區塊而開始，其中$r<\log_2(n)$。所有區塊能以相同方式處理，所以該方法可在第一區塊證明。第一r個加總能以toplitz矩陣與一向量的乘積表示。然後，考慮rx(n+r-1)toplitz矩陣：

五、發明說明（ 38 ）

$$A=\begin{bmatrix} u_1 & u_2 & u_3 & u_4 & \cdots & u_n & 0 & \cdots & 0 & 0 \\ 0 & u_1 & u_2 & u_3 & \cdots & \cdots & u_n & 0 & \cdots & \cdots \\ 0 & 0 & u_1 & u_2 & \cdots & \cdots & \cdots & u_n & 0 & \cdots \\ 0 & 0 & 0 & u_1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & & & & \cdots & & & & \cdots & \cdots \\ \cdots & & & & \cdots & & & & 0 & 0 \\ \cdots & & & & \cdots & & & & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & 0 & u_1 & u_2 & \cdots & \cdots & u_n \end{bmatrix}$$

而且，(n+r-1)維矩陣向量如下示：

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ \cdots \\ \cdots \\ x_{n+r-1} \end{bmatrix}$$

然後，第一r個加總可由下示向量提供：

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ \cdots \\ \cdots \\ y_r \end{bmatrix}$$

其中：　　y=A•x。

本發明較佳具體實施例的基本觀念是要重組特定問題，以致於上述U方法將可適用。根據他們順序，讓$v_1, v_2, \ldots\ldots, v_{n+r-1}$是矩陣A的欄。可看出，所有下列"中間"欄向量是U向量：

五、發明說明（ 39 ）

$$v_r = \begin{bmatrix} u_r \\ u_{r-1} \\ \cdot \\ \cdot \\ \cdot \\ u_1 \end{bmatrix}, \quad v_{r+1} = \begin{bmatrix} u_{r+1} \\ u_r \\ \cdot \\ \cdot \\ \cdot \\ u_2 \end{bmatrix}, \ldots\ldots\ldots\ldots\ldots\ldots, v_n = \begin{bmatrix} u_n \\ u_{n-1} \\ \cdot \\ \cdot \\ \cdot \\ u_{n-r+1} \end{bmatrix}$$

同時注意下列"匹配"對"旁邊"欄向量的加總及減算亦是U向量：

$$v_1 + v_{n+1} = \begin{bmatrix} u_1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ u_n \\ \cdot \\ \cdot \\ \cdot \\ u_{n-r+2} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_n \\ u_{n-1} \\ \cdot \\ \cdot \\ u_{n-r+2} \end{bmatrix}, \qquad v_1 - v_{n+1} = \begin{bmatrix} u_1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ u_n \\ \cdot \\ \cdot \\ \cdot \\ u_{n-r+2} \end{bmatrix} =$$

$$\begin{bmatrix} u_1 \\ -u_n \\ -u_{n-1} \\ \cdot \\ \cdot \\ -u_{n-r+2} \end{bmatrix}$$

$$v_2 + v_{n+2} = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_n \\ \cdot \\ \cdot \\ u_{n-r+3} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_n \\ \cdot \\ \cdot \\ u_{n-r+3} \end{bmatrix}, \qquad v_2 - v_{n+2} = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ u_n \\ \cdot \\ \cdot \\ u_{n-r+3} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ -u_n \\ \cdot \\ \cdot \\ -u_{n-r+3} \end{bmatrix}$$

.....

.....

五、發明說明（ 40 ）

$$v_{r-1} + v_{n+r-1} = \begin{bmatrix} u_{r-1} \\ u_{r-2} \\ \cdot \\ \cdot \\ u_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ u_n \end{bmatrix} = \begin{bmatrix} u_{r-1} \\ u_{r-2} \\ \cdot \\ \cdot \\ u_1 \cdot \\ u_n \end{bmatrix}, \qquad v_{r-1} - v_{n+r-1} = \begin{bmatrix} u_{r-1} \\ u_{r-2} \\ \cdot \\ \cdot \\ u_1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ u_n \end{bmatrix} = \begin{bmatrix} u_{r-1} \\ u_{r-2} \\ \cdot \\ \cdot \\ u_1 \cdot \\ -u_n \end{bmatrix}$$

在工作必要初步準備之後，該方法便可引用。可根據上述而重新配置：

$$A \cdot x = \square_{1 \le j \le n+r-1} \; x_j v_j$$

$$= \square_{1 \le j \le r-1} \; x_j v_j + x_{n+j} v_{n+j} + \square_{r \le j \le n} \; x_j v_j$$

其次，本發明可使用如果x、y是數量且v、u是向量的規則，然後：

$xv + yu = \frac{1}{2}(x+y)(v+u) + \frac{1}{2}(x-y)(v-u)$. 因此：

$$A \cdot x = \square_{1 \le j \le r-1} \; \frac{1}{2}(x_j + x_{n+j})(v_j + v_{n+j}) + \frac{1}{2}(x_j - x_{n+j})(v_j - v_{n+j}) + \square_{r \le j \le n} \; x_j v_j$$

此處理是2r-2個加算，而且達成的形式是一rx(n+r-1)U矩陣及一n+r-1向量的乘積。此在上述所有向量皆如此：

$v_r, v_{r+1}, \ldots, v_n, v_1 + v_{n+1}, \ldots\ldots, v_{r-1} + v_{n-r-1}, v_1 - v_{n+1}, \ldots\ldots, v_{r-1} - v_{n+r-1}$

是U向量。如此其餘計算可藉由U方法達成。除了實質加算/減算之外，U形式修改的所有觀點是：$x_j \pm x_{n+j}$，其是在輸入向量達成之前以一"每次工作"達成。

在最壞情況的加算次數可由下式表示：

五、發明說明（ 41 ）

$$s_t(n,r)=s(n+r-1,r)+2r-2=n+3r-3+s(r)$$

在加總m可被計算及$m>\log_2(n)$的一般設定上，在最壞情況的加算次數範圍是$(1+\square)(n+3\log_2(n))\cdot m/\log_2(n)$，其中$1>\square>0$，而且當m和n是無窮大時，□會是零。

根據本發明的另外一較佳具體實施例，GEM的上述方法元件可整合。在此情況，一些"旁邊"欄可保持不由結合旁邊欄的第一階段處理。

(0,1,-1)矩陣

Toplitz矩陣是一部分的更一般類型矩陣：這些矩陣的登錄是0、1或-1。此矩陣在此稱為(0,1,-1)矩陣。與上述toplits方法有關觀念的一較大版本現將發展。假設A是維矩陣rxn的(0,1,-1)矩陣，而且x是n維矩陣輸入向量，而且它想要計算乘積：A•x。此乘積能以A欄乘以對應x元件之一加總。因此，可由vj(對於每個j而言)表示成A的欄j，然後：

$$A\cdot x=x_1v_1+x_2v_2+\ldots..+x_nv_n$$

一些、或者或許A的所有欄包含0登錄，為了表示簡化的緣故，可假設索引的配置可以是該等k欄$v_1,v_2,\ldots..,v_k$之中每一者包含一零登錄，而且所有其餘n-k欄(如存在)$v_{k+1}$, $v_{k+2},\ldots..,v_n$是U向量(即是，沒有0登錄)。很清楚，該等"混合"向量：$v_1, v_2, \ldots..., v_k$之中每一者是2個U向量的平均。因此，對於每個$1\le j\le k$而言，存在著2個U向量$u_j,w_j$，以致於：$v_j=\frac{1}{2}(u_j+w_j)$。本發明的較佳具體實施例可發現向量$u_1$, $w_1$ $\ldots..., u_k, w_k$是初步每個矩陣一次的準備。因此，藉由上述：

五、發明說明（ 42 ）

$$A \bullet x = \frac{1}{2} x_1(u_1+w_1)+\frac{1}{2} x_2(u_2+w_2)+\ldots\ldots\ldots\ldots+\frac{1}{2} x_k(u_k+w_k)+x_{k+1}v_{k+1+}\ldots\ldots\ldots\ldots+x_n v_n$$

然而，隨著角括弧的開始，我們發現一rx(n+k)U矩陣與一(n+k)向量的一乘積。根據本發明的一較佳具體實施例，此工作將可藉由本發明的上述U觀面完成。本發明的上述toplitz觀面是此廣泛觀念的一特殊情況。

個別應用範例

搜尋器通常是由一Toplitz矩陣與一資料向量的乘積表示。此在CDMA及寬帶CDMA皆如此。

實數矩陣

根據本發明的另一較佳具體實施例，計算運算的數量可藉由使用分配算術而於任何實數線性轉換減少。在本文中，它稱為"實數矩陣方法"。線性轉換是由登錄是實數的一矩陣所表示，而且具有一固定數量的二進位數字，其中該等登錄不必然是整數。矩陣可分解成有二進位矩陣與二進位係數的加總。在下一階段中，本發明的二進位具體實施例將可應用。為了要介紹此方法，可考慮下列範例。範例：考慮下列3×8 U矩陣A，其具有(為了此範例的簡化緣故)整數登錄，而且以十進位的基礎可寫成如下示：

$$A=\begin{bmatrix} 2 & 1 & 5 & 4 & 3 & 0 & 4 & 5 \\ 5 & 0 & 4 & 1 & 4 & 7 & 1 & 2 \\ 2 & 7 & 3 & 1 & 4 & 5 & 0 & 3 \end{bmatrix}$$

8維矩陣向量的輸入是無下所示提供：

五、發明說明（ 43 ）

$$x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

它想要計算3維矩陣輸出向量：

$$y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}=A\bullet x.$$

既然處理器可在二進位基礎正常工作，所以矩陣A將可在此基礎上表示，如此它可藉由下式提供：

$$A=\begin{bmatrix} 010 & 001 & 101 & 100 & 011 & 000 & 100 & 101 \\ 101 & 000 & 100 & 001 & 100 & 111 & 001 & 010 \\ 010 & 111 & 011 & 001 & 100 & 101 & 000 & 011 \end{bmatrix}$$

此表示建議使用分配算術來表3個二進位矩陣加總的矩陣的可能性：

$$A = \square\bullet A[0] + \square\bullet A[1] + \square\bullet A[2]$$

其中：

$$A[0]=\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

五、發明說明（ 44 ）

$$A[1] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A[2] = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

如此由本發明的此具體實施例採用的第一步驟是二進位矩陣A[0]、A[1]、A[2]的結構，其可根據他們的重要性而包含A的登錄位元。在考慮下的較佳具體實施例將認為是一連串相等：

$$A \cdot x = \square \cdot A[0] \cdot x + \square \cdot A[1] \cdot x + \square \cdot A[2] \cdot x$$

下一步驟是組成由A[0]、A[1]、A[2]的水平鏈形成的24×3二進位矩陣A*：

$$A^* = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

而且，亦形成24維矩陣欄向量x*，其包含向量x的3個複製品，其每個具由從上述加總推論的對應二進位加權。此可在輸入向量開始達成之前完成。

五、發明說明（ 45 ）

$$x^* = \begin{bmatrix} 2^0 x_1 \\ \cdot \\ \cdot \\ 2^0 x_8 \\ 2^1 x_1 \\ \cdot \\ \cdot \\ 2^1 x_8 \\ 2^2 x_1 \\ \cdot \\ \cdot \\ 2^2 x_8 \end{bmatrix}$$

注意，乘以$2^1$只需要索引運算。此具體實施例的關鍵是推斷：

$$y=A\bullet x=A^*\bullet x^*$$

如此，下列藉由0-1方法的$A^*\bullet x^*$計算將會發生。因此，下一工作是要歸納及加總共同非零欄係數，以建立新係數，而且藉此減少矩陣大小。

因此，它可保持下式：

$$y = w_1\begin{bmatrix}1\\0\\0\end{bmatrix} + w_2\begin{bmatrix}0\\1\\0\end{bmatrix} + w_3\begin{bmatrix}1\\1\\0\end{bmatrix} + w_4\begin{bmatrix}0\\0\\1\end{bmatrix} + w_5\begin{bmatrix}1\\0\\1\end{bmatrix} + w_6\begin{bmatrix}0\\1\\1\end{bmatrix}$$

其中：

$w_1=2^2\cdot x_7+2^2\cdot x_8+2^1\cdot x_4+2^1\cdot x_5+2^0\cdot x_5$

$w_2=2^2\cdot x_1+2^1\cdot x_6+2^0\cdot x_1+2^0\cdot x_7$

$w_3=2^2\cdot x_3$

五、發明說明（ 46 ）

$w_4=2^2\cdot x_2+2^1\cdot x_2+2^1\cdot x_3$

$w_5=2^1\cdot x_5+2^0\cdot x_2+2^0\cdot x_1+2^0\cdot x_3+2^0\cdot x_8$

$w_6=2^2\cdot x_1+2^1\cdot x_6+2^0\cdot x_1+2^0\cdot x_7$

此可使用16個加算達成。其次，結果矩陣的一列分割將會發生。因此：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = w_1\begin{bmatrix} 1 \\ 0 \end{bmatrix} + w_2\begin{bmatrix} 0 \\ 1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix} + w_5\begin{bmatrix} 1 \\ 0 \end{bmatrix} + w_6\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

及：

$y_3=w_1+w_3+w_5$

現在，加總第一方程式的共同欄係數將可產生下列：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = (w_2+w_5)\begin{bmatrix} 1 \\ 0 \end{bmatrix} + (w_2+w_6)\begin{bmatrix} 0 \\ 1 \end{bmatrix} + w_3\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

此步驟使用2個加算。其餘是使用2個更多加算達成。在此範例中，整個20個加法運算需要轉換計算。傳統方法的使用將需要相等的29個加法運算，其中藉由"相等"，我們意謂額外包含乘積運算亦可考慮。

很顯然，此不是傑出節省的一範例。它是相當容易說明本發明的此具體實施例觀念。然而，當矩陣(對於每個登錄的數字維矩陣與數量)的參數較大時，實質節省可藉由本發明完成。

大體上，在考慮下，本發明的較佳具體實施例係有關於一實值、無限制線性轉換的一有效計算。代表轉換的rxm矩陣A可如下示提供：

五、發明說明（ 47 ）

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_{r1} & & & & & & a_m \end{bmatrix}$$

其中矩陣的登錄是實數。它是要計算A•x，其中x是實數或複數數量登錄之一n維矩陣向量，其寫出如下示：

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

假設矩陣A的登錄是以二進位基礎寫出，而且在點的前後有固定數量的數字。本發明的兩較佳具體實施例可選擇性解決提供，而且在他們之間的選擇是因轉換矩陣的結構而定。第一稱為0-1分解，第二稱為U分解。

假設矩陣A的登錄是範圍$2^m1$，而且在適當正整數$m_1$和$m_2$的點後具有$m_2$數字的精確性。既然由一處理器遇到的每個數量在點的前後具有一有限數量的數字，所以此假設並未限制本發明的範圍。讓$m=m_1+m_2+1$。分配算術將可用來將乘積A•x分成由上述範例所處理類型的m個二進位乘積之一加總。

首先，本發明是在矩陣的登錄是非負數，且分解是0-1基

五、發明說明（ 48 ）

礎的情況中描述。範圍$2^{mt}$的一實目t及具有超過點的$m_2$數字能以下列方式表示：

$$t = \square_{-m_2 \le k \le m_1} t_k \cdot 2^k$$

其中，每個$t_k$是0或1。此是標準二進位表示法。藉由一典型分配算術引數，A能以下列方式分解成m rxn 0-1二進位矩陣之一加總：

$$A = \square_{-m_2 \le k \le m_1} 2^k \cdot A[k]$$

在此加總中，$A[m_1]$是最高有效位元(MSB)的0-1矩陣，$A[-m_2]$是最低有效位元(LSB)的0-1矩陣。大體上，矩陣A[k]之中每一者是0-1矩陣，其登錄是由A登錄之k的有效位元所組成，其中每個位元是放置在它對應登錄。藉由分配規則，它會如下所示：

$$A \cdot x = \square_{-m_2 \le k \le m_1} 2^k \cdot A[k] \cdot x.$$

此是本發明的目前具體實施例的基礎。其次，讓A*是由他們加總出現順序的上述加總的二進位矩陣水平極限所組成的rx(m•n)0-1矩陣。然後，

$$A^* = [A[-m_2], \ldots\ldots\ldots\ldots, A[0], \ldots\ldots\ldots\ldots, A[m_1]]$$

此可在輸入向量達成開始之前每次於任何特定矩陣達成。此外，讓m•n維矩陣欄x*可由下列提供：

五、發明說明（ 49 ）

$$x^{*}=\begin{bmatrix} 2^{-m_2}x \\ 2^{-m_2-1}x \\ \vdots \\ 2^{0}x \\ \vdots \\ 2^{m_1}x \end{bmatrix}$$

在考慮下，具體實施例強調主要可看出：

$$A \cdot x = A^{*} \cdot x^{*}$$

後者是rx(m•n)0-1矩陣與一(m+1)•n維矩陣向量的一乘積，該乘積是由最初向量的偏移m個複製品所組成。

乘積A*•x*的計算可使用本發明的0-1二進位具體實施例達成。既然每個乘以2的一整數乘冪可藉著將位元移位實施，所以非常小的加算複雜度便會增加。

本發明的此具體實施例可明顯減少乘積的複雜度。讓：

$$l = \log_2(m \cdot n) = \log_2(m) + \log_2(n)$$

而且，讓C(A)是本發明所需的加算次數，以計算A*•x*。如果l≥r，那麼：

$$C(A) < m \cdot n + 2^{r} + 2^{r/2+1} - r$$

特別是在此情況中，我們可看出實際範圍是：

$$C(A) < 2m \cdot n$$

當我們假設l≥r，它保存：

五、發明說明（ 50 ）

$$C(A)<(1+\square)\cdot m\cdot r\cdot n/l$$

其中1>□>0，而且當m•n與r是無窮大時，□便會是零。

大體上，計算乘積A•x的傳統(布魯特斯力量)先前技藝需要相等的n•r•m/2個加算，其中包含乘積運算的加算可考慮。

在一些情況中，特別是當r=1、或當r是相當小時，上述具體實施例的另一版本是較佳，以允許進一步節省注意，在情況r=1，問題可減少到兩向量的數量乘積。此本身在科學與技術方面是一相當普遍的計算，而且它的有效性能可使用。根據此變化，矩陣A**可藉著使在一垂直序列中的矩陣$A[-m_2]$, $A[-m_2]$,....., A[0],.....,$A[m_1]$成連鎖狀形成。如此，它可藉由下式提供的r•(m+1)xn 0-1矩陣：

$$A^{**}=\begin{bmatrix} A[-m_2] \\ \cdot \\ \cdot \\ \cdot \\ A[0] \\ \cdot \\ \cdot \\ \cdot \\ A[m_1] \end{bmatrix}$$

然後，A**•x的計算可藉由0-1矩陣的本發明較佳具體實施例達成。乘積A**•x包含所有乘積：$A[-m_2]$•x,.....,A[0]•x,.....,$A[m_1]$•x。因此，使用二進位乘法的移位，想要的結果便可藉由執行加總達成：

$$y=\square_{-m_2\le k\le m_1} 2^k\cdot A[k]\cdot x$$

五、發明說明（ 51 ）

此可推論具負數登錄的矩陣部分。

當矩陣A是具有兩符號登錄的實數時，那麼矩陣A可表示成具負數登錄的兩矩陣減算：$A=A_1-A_2$。遵循此表示，在討論下的本發明觀點可藉由先分別計算該等乘積$y_1=A_1 \cdot x$和$y_2=A_2 \cdot x$、或上述方法的一組合形式而實施$y=A \cdot x$的計算。然後，最後步驟是減法實施：$y=y_1-y_2$。當分解二進位矩陣時，有關一實數的本發明的一較佳的具體實施例的上述0-1二進位選擇是特別有效：$A[-m_2]$, $A[-m_2+1]$,.....,A[0],.....$A[m_j]$是相當稀疏。此可以是各種不同大小的一連串A登錄，而且在點後面具有非一致性數量的二進位數字。在此情況，上述一致性數位格式所需的零填補會造成一較高位準的稀疏。

根據U二進位分配算術，本發明的較佳具體實施例的另一形式將在下文描述。本發明的此形式具有可更適於具有兩符號及根據較快U方法登錄矩陣的優點。實際上，當矩陣登錄的大小相同及準確性時，它可比上述0-1版本更有效。下列方法的主要特徵是類似0-1方法。

下列觀察對於本發明的的處理描述是需要的。範圍$2^{m1}$及具有超過點$m_2$數位的一實數t能以下列方法的U二進位加總表示：

$$t = \square_{-m_2-1 \le k \le m_1-1} s_k \cdot 2^k + s_{m_1} \cdot (2^{m1} - 2^{-m2-1}).$$

其中對於$-m_2-1 \le k \le m_1-1$而言，所有$s_k$是±1，而且當t是非負數時，$s_{m1}=1$，且當t是負數時，$s_{m1}=-1$。

五、發明說明（ 52 ）

因此，具兩符號登錄的一rxn實數矩陣A能以下列方法分解成U矩陣的加總：

$$A = \square_{-m_2-1 \le k \le m_1-1} 2^k \cdot A[k] + (2^{m1} - 2^{-m2-1}) \cdot A[m_1].$$

其中該等矩陣A[k]之中每一者是rxn U矩陣。

讓A*是rx((m+1)•n)U矩陣：

$$A^* = [A[-m_2 - 1],\ A[-m_2],\ldots\ldots,A[0],\ldots\ldots,A[m_1 - 1],\ A[m_1]]$$

此矩陣是在輸入資料到達之前，由每矩陣的一時間工作組成。

此外，(m+1)•n維矩陣欄向量x*是由下列定義：

$$x^* = \begin{bmatrix} 2^{-m_2-1}x \\ 2^{-m_2}x \\ \cdot \\ \cdot \\ 2^0 x \\ \cdot \\ \cdot \\ 2^{m_1-1}x \\ (2^{m_1} - 2^{-m_2-1}) \cdot x \end{bmatrix}$$

它保持；

$$A \cdot x = A^* \cdot x^*$$

此是一rx((m+1)•n)U矩陣乘以一A(m+1)•n維矩陣向量的一乘積。此乘積的計算可由本發明的U矩陣具體實施例的應用達成。

五、發明說明（ 53 ）

在屬於0-1二進位分解的上述方法中，亦存在r=1或小r的垂直版本。它是完全類似先前的描述，而且沒有理由重複細節。

讓l=log((m+1)•n)=log(m+1)+log(n)。它可保持l≥r，執行上述方法的所需的加算次數的範圍如下：

$$C(A)<(m+1)\bullet n + 2^{r-1} + 2^{r/2+1} - r$$

如前述，C(A)是定義成完成A•x的上述U具體實施例所需的加算次數。一般而言，它可保持，

$$C(A)<(1+\square)\bullet(m+1)\bullet r\bullet n/l$$

其中1>□>0，而且當(m+1)•n和r是無窮大時，□會是零。

工業應用範例

線性轉換普遍使用在技術與科學的每個領域。在通訊技術的本發明的實數矩陣觀點應用包括多使用者偵測器(MUD)矩陣(例如解除關聯器或最小均方誤差(MMSE)矩陣)與一解展佈器的輸出向量乘積。它亦可使用在最小平方的計算。有限衝動響應(FIR)濾波器，其中非連續傅利葉轉換(DFT)可完全或部分計算。尤其是部分DFT的FIR濾波器及FFT不是有效的小及中等大小的濾波器是範例，其中本發明的此特徵是可適用。非連續餘弦轉換DCT是本發明改善計算的另一類型線性轉換。當它只是部分計算、或當它的大小不太大，以致於較大維矩陣的快速演算法不是很有效時，此便非常特別有效。

在例如處理採用FIR的處理電路的一些數位信號處理應

五、發明說明（ 54 ）

用中，兩相對長向量的關聯性是需要。該等向量之中一者是表示可在一第二向量上運算的一FIR濾波器接頭，其表示一輸入應該被過濾。由部分捲積組成的過濾操作可藉由一Toplits矩陣與一向量的乘積表示。此可藉由本發明的實數矩陣觀點而有效達成。

複數矩陣

範例：假設它想要計算下列加總：

$y_1=(1+j)\cdot x_1+(1-j)\cdot x_2+(-1-j)\cdot x_3+(-1+j)\cdot x_4+(1-j)\cdot x_5+(-1+j)\cdot x_6$

$y_2=(-1+j)\cdot x_1+(1+j)\cdot x_2+(1+j)\cdot x_3-(-1-j)\cdot x_4+(-1-j)\cdot x_5+(1-j)\cdot x_6$

$y_3=(1-j)\cdot x_1+(-1-j)\cdot x_2+(-1-j)\cdot x_3+(1+j)\cdot x_4+(-1+j)\cdot x_5+(1+j)\cdot x_6$

其中輸入數量$x_1,x_2,\ldots\ldots\ldots\ldots\ldots,x_6$是複數。傳統先前技藝方法將需要66個實際加算。此可看出當我們考慮時，$(\pm1\pm j)$的一因素與一複數的乘積需要2個實際加算，且2個複數的一加算需要2個實際加算時。用以執行此計算的本發明較佳具體實施例之兩主要選項可提供。本發明的第一較佳具體實施例可稱為相位旋轉+GEM，它係使用下式：

$\frac{1}{2}(1+j)\cdot(1+j)=j$

$\frac{1}{2}(1+j)\cdot(1-j)=1$

$\frac{1}{2}(1+j)\cdot(-1+j)=-1$

$\frac{1}{2}(1+j)\cdot(-1-j)=-j$

因此，藉著將所有上述加總乘以$\frac{1}{2}(1+j)$，此動作在此稱為相位旋轉，我們可從組$\{1,-1,j,-j\}$取得下列係數：

五、發明說明（ 55 ）

$$\frac{1}{2}(1+j)\cdot y_1=j\cdot x_1+1\cdot x_2+(-j)\cdot x_3+(-1)\cdot x_4+1\cdot x_5+(-1)\cdot x_6$$

$$\frac{1}{2}(1+j)\cdot y_2=(-1)\cdot x_1+j\cdot x_2+j\cdot x_3+(-j)\cdot x_4+(-j)\cdot x_5+1\cdot x_6$$

$$\frac{1}{2}(1+j)\cdot y_3=1\cdot x_1+(-j)\cdot x_2+(-j)\cdot x_3+j\cdot x_4+(-1)\cdot x_5+j\cdot x_6$$

根據本發明的一較佳具體實施例，這些加總可藉由GEM計算。由於此範例的規模小，增益在與傳統方法相比較的情況中是最低的，但是它實質是在較大維矩陣。當維矩陣如同在此範例較小時，在相位旋轉步驟之後，計算的其他更傳統方法亦可應用。最後，每個加總的結果是乘以(1-j)，以獲得想要的結果。注意，乘以j或(-1)是需要不太多的時間量與能量的"組織"運算。

較佳具體實施例的第二選項稱為複數U方法。它可藉由下列方法解開每個係數的角括弧而將加總分解成多項：

$$y_1=x_1+(jx_1)+x_2-(jx_2)-x_3-(jx_3)-x_4+(jx_4)+x_5-(jx_5)-x_6+(jx_6)$$

$$y_2=-x_1+(jx_1)+x_2+(jx_2)+x_{3+}(jx_3)-x_4-(jx_4)-x_5-(jx_5)+x_6-(jx_6)y_2$$

$$y_3=x_1-(jx_1)-x_2-(jx_2)-x_3-(jx_3)+x_4+(jx_4)-x_5+(jx_5)+x_6+(jx_6)$$

其餘可藉由應用U方法達成。既然s(12,3)=12+3=15，所以此藉著s(n,r)的上述表格而需要30個實際加算，其中此是複數加算的次數。

在藉由上述範例證明基本原理之後，它目前適於提供一般情況的詳細描述。為了描述與複數矩陣有關的本發明較佳具體實施例，可考慮當作係數使用的組：$U_1=\{1,-1,j,-j\}$和$U_2=\{1+j,1-j,-1+j,-1-j\}$。一$U_1$向量或$U_1$矩陣具有在$U_1$的登

五、發明說明（ 56 ）

錄。同樣地，$U_2$向量或$U_2$矩陣具有在$U_2$的登錄。此矩陣與向量是無線電應用是普遍的。它應該考慮一$U_2$數目與一向量數目的乘積需要2個實際加算，而一$U_1$數目與一複數的乘積包括一相當小的複雜度。

第一計算問題的解決如下示：假設提供一rxn $U_2$矩陣A及一n維矩陣複數欄輸入向量x，而且它想要計算乘積y=A•x。數量乘積情況r=1可包括。此計算的兩主要方法可提供。當適當時，每個將會較佳。隨著略微修改，當資料向量是實數時，相同處理便可適用。本發明的第一相位旋轉+GEM較佳具體實施例可引用。讓

$$B=\frac{1}{2}(1+j)\cdot A \text{和} z=\frac{1}{2}(1+j)y$$

然後，B是一rxn $U_1$矩陣及z= B•x。然後，乘積z=B•x可藉由GEM計算。只要z被計算，輸出向量y便可由下列乘積找到：y=(1-j)•z。在從初始相位旋轉步驟產生傳統方法的增益可節省2r•(n-1)加算。此增益可甚至存在r=1的情況，即是，數量乘積的情況。此外，增益是從GEM應用產生。

本發明的第二較佳具體實施例稱為U複數方法。第一件事物是要將A表示成加總：$A=A_1+jA_2$，其中$A_1$和$A_2$是U矩陣。然後，考慮品質：$A\cdot x=A_1\cdot x+jA_2\cdot x$。此等式表示A•x可藉由rx2n U矩陣$A^*=[A_1, A_2]$與2n複數維矩陣欄向量的乘積而計算：$x^*=\begin{bmatrix} x \\ jx \end{bmatrix}$。此是以下列等式表示：$A\cdot x=A^*\cdot x^*$。現在，乘積$A^*\cdot x^*$將可藉由U方法計算。

在本發明的具體實施例中有另一選擇，當r很小時，此是

五、發明說明（ 57 ）

合理：讓A**=$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$，此是2rxn U矩陣。然後，將U方法用來計算乘積：A**•x。此可計算兩乘積$y_1=A_1•x$和$y_2=A_2•x$。該處理可使下列加算完成：$y=y_1+y_2$。

上述問題的變化會在一些應用上發生，包括在CDMA中代表PN相關器的toplitz矩陣表示。在此設定中，矩陣亦具有零登錄。因此，讓：$U'_1=\{0,1,-1,j,-j\}$和$U'_2=\{0,1+j,1-j,-1+j,-1-j\}$。一$U'_1$向量或$U'_1$矩陣具有在$U'_1$的登錄。同樣地，$U'_2$向量或$U'_2$矩陣具有在$U'_2$的登錄。一rxn $U'_2$矩陣A可提供及一n維矩陣複數欄輸入向量x。該目標是要計算乘積y=A•x。在數量乘積r=1的情況是包括在內。

本發明的相位旋轉+GEM較佳具體實施例將先討論。讓

$$B=\frac{1}{2}(1+j)•A \text{和} z=\frac{1}{2}(1+j)y$$

那麼B是一rxn $U_1'$矩陣及z= B•x。其次，乘積z=B•x可藉由GEM的應用計算，或許，當維矩陣較低時，可藉由傳統方法計算。最後，只要z計算，輸出向量y可藉由下列乘積找到：y=(1-j)•z。

根據本發明的另外一較佳具體實施例，A是先表示一加總：$A=A_1+jA_2$其中$A_1$和$A_2$的(0, 1, -1)矩陣，可能是Toplitz。然後，藉由等式：$A•x=A_1•x+jA_2•x$，乘積A•x可藉由rx2n(0, 1, -1)矩陣$A^*=[A_1, A_2]$與2n複數維矩陣欄向量：$x^*=\begin{bmatrix} x \\ jx \end{bmatrix}$的乘積計算。最後，乘積A*•x*將可藉由Toplitz或藉由本發明的一般(0,1-1)觀點計算。當r很小時，反映非Toplitz類似物的本發明的另外一(選擇性)較佳具體實施例是有效：讓

五、發明說明（ 58 ）

$A^{**}=\begin{bmatrix}A_1\\A_2\end{bmatrix}$，此是2rxn(0,1,-1)矩陣。然後應用本發明的(0,1,-1)觀點，以計算乘積：$A^{**}\cdot x$。此可計算乘積$y_1=A_1\cdot x$和$y_2=A_2\cdot x$。最後，該處理可使用下列加總完成：$y=y_1+j\cdot y_2$。

複數現可推論一方法可用於計算一般複數rxn矩陣$A\in C^{rxn}$與一實數或複數n維矩陣向量x的乘積。根據本發明的較佳具體實施例，先將A表示成一加總：$A=A_1+jA_2$，其中$A_1$和$A_2$是實數矩陣。其次，透過等式：$A^*x=A_1^*x-jA_2^*x$。既然$A\cdot x=A^*\cdot x^*$，所以$A^*x$可藉由rx2n實數矩陣$A^*=[A_1,A_2]$與2n維矩陣欄向量$x^*=\begin{bmatrix}x\\jx\end{bmatrix}$的乘積而計算。最後，乘積$A^*\cdot x^*$可藉由實數矩陣方法計算。

根據本發明的另外一(選擇性)較佳的具體實施例，讓$A^{**}=\begin{bmatrix}A_1\\A_2\end{bmatrix}$。此是一2rxn實數矩陣。然後，應用實數矩陣方法，以計算乘積：$A^{**}\cdot x$。此可計算乘積$y_1=A_1\cdot x$和$y_2=A_2\cdot x$。最後，該處理可使用下列加總完成：$y=y_1+j\cdot y_2$。

最後，具$U_2$係數的一toplitz矩陣與一向量的乘積可藉由上述toplitz技術的應用達成。

Toplitz矩陣、(0,1,-1)矩陣、與複數矩陣技術的工業應用範例

IS-95搜尋器：IS-95 CDMA系統在小地理位置中允許許多不同基地台，以便同時使用可供將資料傳送給行動接收器的相同頻譜區段。來自不同基地台的資料可不同的方式是經由可用來展佈傳輸資料的PN序列。每個是根據PN序列的一不同相位。在行動接收器中的搜尋器機構工作可藉著

五、發明說明（ 59 ）

將他們的PN相位排列而識別由環境基地台所傳送的不同導頻信號。它同時可應用於來自相同基地台的數個多重路徑信號(表示回覆)之間的不同。一類似處理可應用在初始的同步程序。

對於每個假設而言，一搜尋器可使用一區域產生的PN序列而測試接收信號部分關聯性的許多假設。然後，序列可於每個假設偏移，而且關聯性可於固定數量的信號元件(晶片)執行。正常上，搜尋器需要在一特定視窗搜尋所有假設，其中每次序列可移位1。一搜尋器可藉由構成一矩陣A而在一DS-CDMA系統中實施，其列是由上述移位PN子序列所組成。搜尋結果然後能以向量y=A•x提供，其中x是代表在單晶週期上取樣輸入信號的一向量。根據本發明的一較佳具體實施例，有效線性轉換的上述創作演算法可實施，如此可獲得可減少資源消耗的向量y。本發明的許多應用在寬頻帶CDMA的建議標準搜尋器中亦很有用。

多重乘積

本發明的另一較佳具體實施例係有關U矩陣與向量乘積部分加總需要的情況。當不同速率(展開因素)的數個碼同時測試時，此便可能在CDMA通訊應用中發生。研究此具體實施例對於達成本發明先前觀點控制的讀者是有助益的。此將藉由下列提供方法觀念範例而概略說明。然而，沒有合理大小的範例可描述具體實施例的所有觀點。讀者亦可參考本發明第6項的概述。

範例：考慮如下5×8 U矩陣：

五、發明說明（ 60 ）

$$A=\begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \end{bmatrix}$$

及一8維矩陣輸入向量：

$$x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

假設，(在多重乘積術語內)列1和2的展開因素是2，而且列3和4的展開因素是4，而且列5的展開因素是8。此表示在最初兩列中，每兩連續元件可加總，在第三及第四列中，每4個連續元件可加總，而且在第五列中，整個列可加總。慣例是展開因素不會減少，即是每列的展開因素是等於、或大於先前列的展開因素此術語稍後將正確定義。

上述說明下列加總將可計算：

$x_1 + x_2$, $-x_3 + x_4$, $x_5 - x_6$, $-x_7 - x_8$

$-x_1 + x_2$, $-x_3 - x_4$, $x_5 + x_6$, $-x_7 + x_8$

$x_1 - x_2 + x_3 + x_4$, $x_5 - x_6 + x_7 - x_8$

$x_1 + x_2 - x_3 + x_4$, $-x_5 - x_6 + x_7 - x_8$

$-x_1 - x_2 + x_3 + x_4 - x_5 + x_6 - x_7 - x_8$

五、發明說明（ 61 ）

首先，分成4個4×2 U矩陣，其中水平維矩陣係反映最低展開因素。基本上，U方法然後可應用，以證明加算如何經由此新觀點節省。因此，藉由應用去除相等列，只有2個加算需用來計算各列的所有第一加總：

$x_1 + x_2$

$-x_1 + x_2$

$x_1 - x_2$

$x_1 + x_2$

$-x_1 - x_2$

同樣地，只有2個加算需用於各列的第二加總：

$-x_3 + x_4$

$-x_3 - x_4$

$x_3 + x_4$

$-x_3 + x_4$

$x_3 + x_4$

等。因此，整個8個加算需用於此部分。4個額外加算可完成列3和4的必要加總，而且另外4個額外加算可用來計算列5的必要加總。因此，整個16個加算可用於本發明較佳具體實施例的應用。傳統brute力方法的先前技藝於相同工作需要28個加算。本發明目前觀點的環境包括可能大維矩陣的一U矩陣，其中在每列上，相等間隔的子加總在上述範例是需要的。輸入向量是實數或複數。矩陣能以列而細分成數個子矩陣，其中每個子矩陣可由在上述範例中所使用

五、發明說明（ 62 ）

的方法個別單獨計算。在此具體實施例中，一方法可結合，以便從節省加算的觀點找出近似最好細分，因此可減少複雜度。它的工具是根據動態程式化的一額外處理器或裝置，該動態程式化可藉由U方法的表格、範圍及複雜的迴歸公式s(n,r)使用來分析各種不同細分。非常精確的形成對於此具體實施例的發展是需要的。數種新定義可視同預備資料而需要。

讓$v=(v_1,v_2,.....,v_n)$是一向量，而且p是除以n(簡而言之：p|n)的一正整數。定義向量v[p]是藉著將v細分成長度p區段而形成。如此：

$$v[p]=((v_1,v_2,.....,v_p),(v_{p+1},v_{p+2},.....,v_{2p}),.....,(v_{n-p+1},v_{n-p+2},.....,v_n)).$$

區段可依下式表示：

$$v[p,k]=(v_{(k-1)p+1},v_2,.....,v_{kp}) \qquad 其中 1 \le k \le n/p.$$

在本文的整數p是稱為展開因素。可看出，多重向量結構是與一矩陣的結構有關。下一項的多重向量的多重數量乘積是在矩陣乘積與平常數量乘積之間的一相交。採用2個n維矩陣向量$v=(v_1,v_2,...,v_n)$和$w=(w_1,w_2,.....,w_n)$，然後v和w的p多重數量乘積可依下列定義：

$$v\bullet w[p]=(v[p,1]\bullet w[p,1],\ v[p,2]\bullet w[p,2],.....,v[p,n/p]\bullet w[p,n/p])$$

其中內部乘積：$v[p,1]\bullet w[p,1]$、$v[p,2]\bullet w[p,2]$,.....是平常數量乘積。注意，此乘積的結果是一n/p維矩陣向量。

讓A是具列$A_1$,.....,Ar的一rxn矩陣且$p=(p_1,p_2,.....,p_r)$是一

五、發明說明（ 63 ）

正整數向量。對於每個j≤i≤r而言，如果$p_i$除以n，便可說是p除以n。此表示：p|n。現假設p|n。採用一n維矩陣向量x及定義A•x的p多重乘積，A•x[p]可如下示表示向量中的向量：

$$A•x[p] = (A_1•x[p_1], A_2•x[p_2], \ldots\ldots, A_r•x[p_r])$$

本發明的目前具體實施例將可改善在一結構中的乘積計算，此將於下面描述。

一多重乘積系統。一多重乘積系統(或簡稱MP系統)是一結構，其包括整數n、r參數、及一正整數向量$p=(p_1,p_2,\ldots\ldots,p_r)$，以致於：

$$p_1|p_2,\ p_2|p_3,\ p_3|p_4\ldots\ldots\ldots,\ p_{r-1}|p_r\ \text{及}\ p_r|n.$$

整數$p_1,p_2\ldots\ldots,p_r$稱為系統的展開因素。MP系統的參數將可依下列方法表示：

$$P=(r,n,p)$$

現要增加一參數rxn U矩陣A及一n維矩陣實數向量x參數。此目標是要有效計算乘積A•x[p]。整個MP系統將能以下式表示：

$$S=(r,n,p,A,x)$$

當所有整數$p_1,p_2,\ldots\ldots,p_r,n$亦是2的乘冪時，那麼系統便稱為二進位多重乘積系統，或簡稱BMP系統。

M-1方法。本發明的此特徵是將U方法直接應用到MP系統。它是由上述範例表示。事實上，它通常可在近似最佳

五、發明說明（ 64 ）

細分之後運用到一MP系統的子系統，而且此將稍後描述。

提供一MP系統S=(r,n,p,A,x)。該方法是基於將矩陣水平分成與最小展開因素$p_1$有關的子矩陣，每個子矩陣是寬度$p_1$。它是藉由計算向量x的第一$p_1$實數與矩陣A的U係數的加總而開始。此可藉由U二進位方法而於所有列同時完成。然後，它會到下一$p_1$欄執行相同處理。它能以此方式持續，直到所有n變數用完為止。其次，它會到該等列之中每一者，其中$P_i>P_1$，並且能以一平常方式完成加算處理。

在每個子矩陣上的U方法的第一步驟可掃描線條，該等線條是其他線條的一可靠或一無用副本。因此，例如，如果$p_1=2$，那麼充其量有2個加算在每個部分需要，而不管r。大體上，僅有$2^{p_1-1}$列可在每個部分經由U二進位方法考慮。而且，本文想要的一應用是當A是Hadamard。在此情況，每個部分不超過$p_1$個非相等列。如此，另一來源可插入，其包含有多少非相等列可在每個子矩陣中出現的範圍。它將包含在由z(當作表格儲存)表示的一函數，而且是既有的矩陣類型。它的參數是$p_1$和r，而且它可寫成$z(p_1,r)$。例如，當A是Hadamard時，那麼z(4,6)=4、z(8,5)=5，當A是一般U矩陣時，那麼z(4,40)=8。它始終保持$z(p_1,r)\leq\min\{2^{p_1-1},r\}$，而且在A是Hadamard的情況：$z(p_1,r)=\min\{p_1,r\}$。

M-1方法的複雜計算將基於在上述U二進位方法描述上出現之s(r)的表格及迴歸公式。對於每個正整數y而言，設定s'(y)=s(y)+y。亦是：$s'(y)<2^{y-1}+2^{y/2+1}$，而且此不平式是較嚴格。此在下一公式可提供複數大小的一直覺範圍。如此

五、發明說明（ 65 ）

，藉由M-1方法達成的加算次數範圍可經由下列各項表示：

$$C(n,r,p,z) = (n/p_1)(p_1 + s(z(p_1,r))) + (n/p_2)(p_2/p_1 - 1) + (n/p_3)(p_3/p_1 - 1) + \ldots\ldots\ldots\ldots\ldots\ldots + (n/p_r)(p_r/p_1 - 1)$$

$$= n\cdot\left(1 + (s(z(p_1,r)) + r - 1)/p_1 - (1/p_2 + 1/p_3 + \ldots\ldots\ldots\ldots + 1/p_r)\right) =$$

$$= n\cdot\left(1 + s'(z(p_1,r))/p_1 - (1/p_1 + 1/p_2 + 1/p_3 + \ldots\ldots\ldots\ldots + 1/p_r)\right)$$

注意，此公式有些瑣碎但是重要例證。首先是當矩陣具有一列，即是，當r=1時，那麼：

$$C(n,r,p)=n-n/p_1$$

其次是一致性展開因素的情況，即是當$p_1=p_2=\ldots\ldots=p_r$，那麼：

$$C(n,r,p,z)=n\cdot(1+ s(z(p_1,r))/p_1)$$

明顯地，當r相對於$p_1$是相當大時，M-1方法便不是很有效。於此基礎上發展的一較聰明方法是一相當大的踏腳石。此較強方法可藉由水平將矩陣細分及將M-1方法個別應用到每個子矩陣達成。若要使用較少計算找到具有較少加算總數的一細分結構，它對於具有上述公式的下列較短版本是很有用的。如此可定義：

$$C^*(n,r,p,z)=1+s'(z(p_1,r))/p_1$$

多重系統的細分及多重向量的一般觀念。其次，一組織化可經由在水平列將一矩陣細分的支配上達成。此支配將可藉由一細分向量r表示。結果可將最初問題分成相同寬度

五、發明說明（ 66 ）

-n的一些子問題，其每個子問題可經由上述M-1方法解決。該細分稍後可用於創作演算法以使效力最大化。可採用展開因素的r維矩陣整數向量$p=(p_1,p_2,.....,p_r)$、及一rxn U矩陣：

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_{r1} & & & & & & a_{m} \end{bmatrix}$$

而且採用整數k、m其中$1\le k\le m\le r$。首先定義向量p的一片段：

$$p(k,m)=(p_k,.....,p_m)。$$

它只是採用索引k到m的元件。定義矩陣A的一片段；

$$A(k,m)=\begin{bmatrix} a_{k1} & a_{k2} & \cdot & \cdot & \cdot & \cdot & a_{kn} \\ a_{k+1,1} & a_{k+1,2} & \cdot & \cdot & \cdot & \cdot & a_{k+1,n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_{m1} & & & & & & a_{nm} \end{bmatrix}$$

而且，它表示採用索引k到m的列。

其次，考慮一整數向量：r=(r(1),....., r(t+1))，可滿足：

$$k=r(1)<r(2)<.....<r(t)<r(t+1)=m+1，$$

其是將一引數分成數個片段。首先，r是一工具，其能以下列方式將一細分的p建立成向量p[r]的一向量p：

$$p[r]=\left((p_{r(1)},.....,p_{r(2)-1}),(p_{r(2)},.....,p_{r(3)-1}),.....,(p_{r(t)},p_2,.....p_{r(t+1)-1})\right)$$

五、發明說明（ 67 ）

子向量能以下列表示，

$p[r,1]=(p_{r(1)},\ldots\ldots,p_{r(2)-1})$

$p[r,2]=(p_{r(2)},\ldots\ldots,p_{r(3)-1})$

.

.

.

$p[r,t]=(p_{r(t)},\ldots\ldots,p_{r(t+1)-1})$

矩陣A的列細分同樣是依照下列方式的向量r細分：

對於所有整數而言，$1\le q\le t$；$A[r,q]=(a_{ij}：r(q)\le i<r(q+1),1\le j\le n)$

一特定細分的M方法。此片段在一機構的構成中是主要步驟，以找到此具體實施例核心的一低複雜細分機構。假設，我們具有矩陣的一列細分，並且可個別在每個子矩陣上執行M-1方法。目標是要評估總加算次數，所以一較少加算細分將可在下一階段找到。固定MP系統 $S=(r,n,p=(p_1,p_2,\ldots\ldots,p_r),A,x,z)$ 及一細分向量 $r=(r(1),\ldots\ldots,r(t))$，其中 $1\le k=r(1)<r(2)<\ldots\ldots<r(t)<r(t+1)=m+1\le r+1$。

對於$1\le q\le t$而言，$S_q$ MP子系統可依下式提供：

$$S_q=(r(q+1)-r(q),\ n,p[r,q],\ A[r,q],x,z)$$

所有子系統的總加算次數是：

$$C(n,r,r,p,z)=\square_{1\le q\le t}C(n,\ r(q+1)-r(q),\ p[r,q],z)=$$

$$=n\cdot\left(\square_{1\le q\le t}s'\left(z(p_{r(q)},\ (r(q+1)-r(q))\right)/p_{r(q)}-(1/p_k+\ldots\ldots\ldots+1/p_m)+t\right)$$

下一目標是要顯現一有效方法，以找到可將此項減少的

五、發明說明（ 68 ）

細分。此將更有效執行這些計算，而無需在每個階段上計算重複子項加算：$1/p_k+.....+ 1/p_m$及n因素。如此，我們可定義：

$$C^*(n,r,r,p,z) = \square_{1\le q\le t}\ s'(z(p_{r(q)}, (r(q+1) - r(q)))/p_{r(q)} + t$$

一近似最佳的細分。在此階段，我們應該研究細分的性質，以減少由上述項C(n,r,r,p,z)提供的加算次數。首先，我們將一抽象公式提供給加算次數，其中細分在最初問題的一特定分間隔上具有最小的最壞情況範圍。

然後，考慮MP系統$S=(r,n,p=(p_1,p_2,.....,p_r),A,x,z)$及整數k、m其中$1\le k\le m\le n$。定義：

$$h(k,m) = min\{C(n, m - k + 1, r, p(k,m),z) : for\ r=(r(1),.....,r(t+1))$$

其中 $k=r(1)<r(2)<.....<r(t)<r(t+1)=m+1\}$

$$h^*(k,m) = min\{C^*(n, m - k + 1, r, p(k,m),z) : for\ r=(r(1),.....,r(t+1))$$

其中 $k=r(1)<r(2)<.....<r(t)<r(t+1)=m+1\}$

它可保持：

$$h(k,m) = n\bullet(h^*(k,m) + (1/p_k +..... + 1/p_m))$$

下列回歸的公式可保持：

$$h(k,m) = min\{C(n, m-k+1, p(k,m), z),\ min\{h(k,q-1) + h(q,m): for\ all\ k < q \le m\}\}$$

其中min(空集合)=無窮大。

$$h^*(k,m) = min\{C^*(n, m-k+1, p(k,m), z)\ min\{h^*(k,q-1) + h^*(q,m): for\ all\ k < q \le m\}\}$$

五、發明說明（ 69 ）

其中min(空集合)=無窮大。

現在，這些表示可由下列動態碼使用，其中子構造是在k與m之間的間隔。若要減少此碼複雜度，h*可取代在路徑上的h，以發現最好的子構造。此對於最後結果不會有影響。若要在最佳細分中找到第一步驟，我們將計算q，以減少每個k和m的式子h(k,q-1)+h(q,k)，其中1≤k≤m≤n，其是可減少式子h*(k,q-1)+h*(q,k)的相同q。如此，可定義：

q(k,m)=k　　當h*(k,m)=C*(n,r,p(k,m),z)及其它：

q(k,m)=最小q，其中k<q≤m且h*(k,m)=h*(k,q-1)+h*(q,k)

現在，最佳細分碼可形成。

可找到最佳細分的一動態程式創作性程式碼。下列程式碼可接收當作資料的MP系統P=(r,n,p,z)的參數，並且產生當作輸出的細分r，其中M方法可最佳執行。此外，它亦可傳回最佳M方法的複雜項。

若要有效執行，我們需要預先計算及儲存s'(r)的表格。此可藉由迴歸公式計算此表格的一快速程式碼達成：

$$s'(r)=2^{r-1}+s'(r(1))+s'(r(2))$$

最佳細分表(n,r,p,z)

for b是從0到r-1 do

for k是從1到r-b do

m：=k+b

h*(k,m)<----C*(n, m-k+1 , P(k,m),z)

q(k,m)<----k

for q是從k+1到m do

五、發明說明（ 70 ）

d <---- h*(k,q-1)+ h*(q,m)

if d < h*(k,m)then

h*(k,m)<---- d and

q(k,m)<----- q

傳回表h*和q。

目前留下是可從此程序所構成q(k,m)獲得最佳細分向量。此可藉由建立包含最佳向量r元件的組R的下列程式碼達成。

最佳細分向量(n,r,p)

設定：R：=空集合及k:=1 m:=r

Find Vector(k,m)：

if q(k,m)> k then

設定q：=q(k,m)

將q加到組R

Find Vector(q,m)

Find Vector(k,q-1)

傳回組R。

既然最佳細分r可找到，所以M方法將可在此細分上執行。

實施

本發明的下列較佳具體實施例可提供上述方法詳細實施的善創作性演算法。他們允許額外減少記憶體、及執行上述方法所需的能量資源。他們具有可提高上述方法及提供建構裝置所需指令的雙重目標。有關這些實施的一者是轉換矩陣是一般的，而且認為是從具有一特定組登錄的矩陣

五、發明說明（ 71 ）

組隨機選取。另一者是當與欄數量相比較時，列數量要足夠小，以致於上述基本原範圍

列數量<log(欄數量)

可滿足。因此，一些步驟適合在其他情況，例如相等列是否在此冗餘的一檢查。

本文描述的映射是遵守從一位置到下一位置的資料流。每個位置是指定對應在一特定重複上一矩陣的欄。在U矩陣情況，欄的一(-1)元件是對應到位址的1，而且同樣地，欄的1元件是對應到位址的0。一類似對應可於$U_1$矩陣定義。

將描述的實施包括一第一步驟，其中輸入xj信號可加到他們對應欄決定的預設目的地，其中每個xj是乘以一符號及/或2的乘冪。在處理重複中，每當一欄根據本發明分割時，兩分割之中每一者的指定位址是少於或等於特定欄的位址。此允許在一些記憶體位置中儲存的每個資訊在處理及遺失之前傳送送它的目的地。此外，如果一特定欄的該等分割成半之中一者是零，代表此欄的位址將可用於下一重複。這些性質可藉由減少資料移動所設定的資料流映射的新結構達成。在每個重複上，位址數量是保持未觸及的，包括提供的內容可由下一重複使用的所有位址。下列實施的機構可從本發明的上述GEM較佳具體實施例的觀點而更了解。包括數值1的一組有限數量可考慮，而且在本文中，如果一r列的矩陣是由元件屬於組S的正常化r維矩陣欄向量的所有可能非零結構所組成，它便是一完整S-r矩陣，其中每個結構只出現一次，而且正常化慣例是最底部非零

五、發明說明（ 72 ）

元件是1。

觀察下列小維矩陣範例。

矩陣：

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

是一完整{0,1}-2矩陣；

矩陣：

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

是一完整U-2矩陣；

矩陣：

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

是一完整U-3矩陣；

矩陣：

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

是一完整{0,1}-3矩陣；

矩陣：

$$\begin{bmatrix} 1 & -1 & j & -j \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

五、發明說明（ 73 ）

是一完整{1,-1,j,-j}-2矩陣。

該等實施是基於列數量與欄數量相較是相當小的假設，所以所有可能結構的較大比例可在轉換矩陣的欄出現。在此，在所有處理實施中達成的第一重複可減少計算一完整S-r矩陣與一修改向量的乘積。結果，所有更進階的重複可計算一較少維矩陣完整S矩陣與一相對修改向量的乘積。

欄在上述範例矩陣中出現的順序方式係反映這些實施位址設定的另一特性。此是一位址根據將該等欄轉換成一基於|S|位址的一致規則，其中MSB是在向量的底部元件，而且LSB是在向量的頂端元件。

基本結果是在第一重複之後，具有獨具一致性完整轉換矩陣，及一致(且相當明顯)位址編號。此一致性矩陣是因S和r而定，而不是初始轉換矩陣。此可導致其餘處理的一致性，因此允許資料流結構在第一階段之後建立，其中該等資料流結構是與特殊轉換矩陣普遍使用或無關。此以硬體實施及基於本發明的裝置結構是非常有用，此是本章的主要目標。這些實施的額外及連續觀點是要減少所需的讀與寫記憶體配置及適當轉換矩陣的加算次數。結果，處理性能所需的能量及裝置的成本可減少。可看出當程式碼處理具數個輸入x向量的相同初始轉換矩陣時，下列程式碼之中每一者的效率可提高。

最後，若要具有一適當預期，下列實施應該閱讀了解，其中他們是本發明的許多可能有效實施中的一些範例。

五、發明說明（ 74 ）

範例1：0-1二進位矩陣

下列一連串步驟係描述本發明的0-1矩陣觀點實施。資料是由一rxn 0-1矩陣$A=(a_{ij}: 0\cdot i<r, 0\cdot j<n)$及一輸入實數或複數向量$x=(x_0,.....,x_{n-1})$所組成。矩陣A的欄是以$v_0,.....,v_{n-1}$表示。此一連串步驟可計算乘積$y=(y_0,.....,y_{r-1})^T=A\bullet x$。在每個特定k重複上，每個位置包含一實數或複數，其是因向量x而定。配置的讀與寫記憶體包含從標示1到$2^r-1$的$2^r-1$位址，其表示在每個反複處理中的欄。下列定義是本發明較佳具體實施例的一部分描述。

定義

1)對於所有k>0、j>1而言，定義：$l(k,j)=\lceil (j-1)\bullet(r/2^k)\rceil$

2)對於所有$0\bullet m<r$而言，定義$Y_m=2^m$。在處理結束，這些位址包含處理輸出向量$y=(y_0,.....,y_{r-1})^T$的元件，其中$Y_0$是$y_0$位址，$Y_1$是$y_1$的位址等。

3)對於k>0及j>1而言，可定義下列函數$F_{k,j}$, $G_{k,j}$，其可控制從每個位置到下一位置的資料移運。首先設定：$l=l(k,j)$、$m=l(k+1,2j)$、$h=l(k,j+1)-1$。

其次，對於每個整數v>0而言，定義：

$$F_{k,j}(v)=2^m\bullet\lfloor v/2^m\rfloor \qquad \text{[方程式100]}$$

$$G_{k,j}(v)=v \bmod 2^m \qquad \text{[方程式101]}$$

4)對於每個向量$v=(v_0,.....,v_{r-1})\in\{0,1\}^r$而言，定義：

$$\square(v)=\sum_{0\le j<r} 2^j v_j.$$

虛擬碼：

五、發明說明（ 75 ）

1.初始化：將零置於從1到$2^r$-1的每個記憶體

2.第一階段：for aj是從0到n-1 do

if vj≠0 than將$x_j$加到位址v= □($v_j$)

3.主要部分：

for k是從0到⌈log(r)⌉-1 do

for j是從1到$2^k$ do

if l(k,j+l)-1 > l(k,j)then

for p是從1到$2^{l(k,j+1)-l(k,j)}$-1 do

設定v=$2^{l(k,j)}$-p

for(來源)位址v do

if $F_{kj}(v)\neq 0$及$F_{kj}(v)\neq v$ then

將(來源)位址v的值加到(目的地)位址$G_{k,j}(v)$的值

及亦將(來源位址)v的值加到(目的地位址)$F_{k,j}(v)$的值

4.獲得輸出：

在此階段，對於所有0·i<r而言，每個位址Yi包含輸出元件$y_i$的值。

說明

本術語的一基本步驟是從稱為來源之一記憶體位置讀取一數值，並且將它加入在稱為目的地的另一記憶體位置中放置的數值。

基於上述虛擬碼的一裝置需要下列基本步驟，以計算上述A•x計算的主要部分：

$$C^{0,1}_{r} \text{ © } 2^{r+1} - 2r - 2$$

此數值只依賴r。

五、發明說明（ 76 ）

如此，A•x的整個計算充其量需要下列整個：

$$C^{0,1}{}_{n,r} \copyright n + C^{0,1}{}_{r}$$

基本步驟。

圖1：0-1二進位矩陣

圖1係描述更新由一裝置所採用一記憶體內容的操作，該裝置可執行上述虛擬碼的主要部分，如此能以一較佳方式實本發明的0-1二進位觀點，而且0-1二進位矩陣包含4個列，在每個重複的每個矩陣欄是以一二進位方式而由一記憶體位址表示。底部(0或1)元件(一水平表示的極右元件)是最高有效值(MSB)，而且最高(0或1)元件(一水平表示的極左元件)是最低有效值(LSB)。在處理結束，位址$Ym=2^m$，0•m•3包含輸出向量$y=(y_0,.....,y_3)^T$的元件，其中$Y_0$是$y_0$的位址，$Y_1$是$y_i$的位址等。箭號係表示採用一位址內容的動作，並且將它傳送，以加入到另一位址的內容。此可如上面虛擬碼的表示而根據重複順序及在每個重複的位址(逐漸增加)順序達成。

範例2：U矩陣

下列步驟序列係描述本發明的U矩陣觀點實施。資料是由一rxn U矩陣$A=(a_{ij}：0•i<r，0•j<n)$及一輸入實數向量$x=(x_0,.....,x_{n-1})$所組成。矩陣A的欄可藉由$w_0,.....,w_{n-1}$表示。此一連串步驟可計算乘積$y=(y_0,.....y_{r-1})^T=A•x$。在每個特定重複上，每個位置包含一或複數，其是因向量x而定。分配的讀及寫記憶體包含從0到$2^{r-1}+r-2$標示的$2^{r-1}+r-1$個位址。下列定義及先前範例的定義於本發明的目前較佳具體實施例

五、發明說明（ 77 ）

是需要的。

定義：

1)對於每個r維矩陣U向量而言，$u=(u_0,.....,u_{r-1})$係定義：

$$Sign(u)=u_{r-1}$$

$$h(u)=(u_0 \cdot u_{r-1},.....,u_{r-1} \cdot u_{r-1}).$$

2)藉由下列定義在雙極二進位組U={1,-1}與邏輯位元二進位組B={0,1}之間的一對應：

$$(-1)'=1$$

$$1'=0$$

因此，對於一r維矩陣U向量而言，$u=(u_0,.....,u_{r-1})$係定義：

$$\pi(u)=\sum_{0\le j<r} 2^j u'_j.$$

3)定義$Y_0=0$，而且對於所有$1 \bullet m \bullet r-1$而言：$Y_m=2^{r-1}+m-1$。在處理結束，這些位址包含輸出向量$y=(y_0,.....,y_{r-1})^T$的元件，其中$Y_0$是$y_0$的位址，$Y_1$是$y_1$的位址等。

4)對於所有k>0, j>1而言，映射$F_{kj}$, $G_{kj}$, $Sign_{k,j}$係依下列定義。讓l=l(k, j)，m=l(k+1, 2j)，h= l(k, j+1)-1。對於每個整數而言，v>0係定義：

$$Sign_{k,j}(v) = 1 - 2 \cdot \lfloor (v \bmod 2^m)/2^{m-1} \rfloor$$

$$F_{k,j}(v) = 2^m \cdot \lfloor v/2^m \rfloor$$

$$G_{k,j}(v) = \begin{cases} v \bmod 2^m & \text{若 } Sign_{k,j}(v) = 1 \\ 2^m - 2^l - (v \bmod 2^m) & \text{若 } Sign_{k,j}(v) = -1 \end{cases}$$

五、發明說明（ 78 ）

虛擬碼：

1.初始化：將零置於從0到$2^{r-1}+r-2$的每個記憶體

2.第一階段：for j是從0到n-1 do

將$Sign(w_j)\cdot x_j$加到位址$\pi(h(w_j))$

3.主要部分：

for k是從0到$\lceil \log(r) \rceil -1$ do

for j是從1到$2^k$ do

if $l(kj+1)-1 > l(k,j)$ then

1)將(來源)位址$Y_{l(k,j)}$的值加到(目的地)位址$Y_{l(kt1,2j)}$的值

2)for p是從1到$2^{t(kj+1)-l(kj)-1}-1$ do

設定$u=2^{l(k,j)}\cdot p$及for來源位址u do

if $G_{kj}(u)=u$ then

將來源位址u的值加到目的地位址的值$Y_{l(k+1,2j)}$

else if $F_{kj}(u)=u$ then

將(來源)位址u的值加到(目的地)位址$Y_{l(k,j)}$的值

else if $G_{kj}(u)=0$將乘以(-1)的(來源)位址u的值加到(目的地)位址$Y_{l(k,j)}$的值及亦將(位址u)的值加到位址$F_{k,j}(u)$的值。

將乘以$Sign_{kj}(u)$的來源位址u的值加到目的地位址$G_{kj}(u)$的值亦將(位址u)的值加到位址$F_{kj}(u)$的值

4.獲得輸出：

對於所有$0\cdot i<r$而言，每個位址Yi目前包含yi的值。

說明

i)在上述實施U設定中的基本步驟是從稱為來源之一記憶體位置讀一數值，將它乘以1或-1的符號值，然後將結果

五、發明說明（ 79 ）

加到稱為目的地的另一記憶體位置中放置的數值。

ii)複雜式可使用下列公式形成。對於0•k•⌈log(r)⌉-1及對於1•j•$2^k$而言，讓：

$$u_{k,j}=2^{l(k,j+1)-l(k,j)-1}$$

$$u_k=\sum_{1\cdot j\cdot 2^k} u_{k,j} = \sum_{1\cdot j\cdot 2^k} 2^{l(k,j+1)-l(k,j)-1}.$$

對於k=⌈log(r)⌉及對於1•j•$2^k$而言，讓：

$$u_{k,j}=\lfloor 2^{l(k,j+1)-l(k,j)-1} \rfloor,$$

那麼：

$$u_k=\sum_{1\cdot j\cdot 2^k} u_{k,j} = r.$$

iii)基於上述虛擬碼的一裝置需要下列基本步驟來實施A•x計算。

(1)對於第一階段n而言，基本步驟是需要的。

(2)對於所有0•k•⌈log(r)⌉-1及1•j•$2^k$而言

$$2\cdot u_{k,j} - u_{k+1,2j} - u_{k+1,2j-1} + 1$$

基本步驟可在步驟(k,j)完成。

(3)因此，對於所有0•k•⌈log(r)⌉-1而言，主要部分的k重複需要下列基本步驟：

$$2^k + 2\cdot u_k - u_{k+1}$$

(4)如此，A•x的上述計算的主要部分需要下列基本步驟：

五、發明說明（ 80 ）

$$C^{u}_{r} \copyright 2u_0+u_1+u_2+.....+u_{\lceil \log(r) \rceil -1} + 2^{\lceil \log(r) \rceil} -1 - r.$$

此數值只依r而定。

(5)如此，A•x的整個計算需要下列總數：

$$C^{u}_{n,r} \copyright n + C^{u}_{r}$$

基本步驟。

圖解2：U矩陣

圖2係描述更新由一裝置所採用記憶體內容的操作，其中該裝置可執行上述虛擬碼的主要部分，如此便能以一較佳方式實施本發明的U二進位觀點及由4列組成的一U二進位矩陣。在每個反複上的每個矩陣能以記憶體中的二進位位址表示，其中欄的(-1)元件是對應在位址的1，而且欄的一1元件是對應位址的0。二進位解譯可被應用，其中底端元件(以一水平表示的極右端元件)是最高有效位元(MSB)，而且在頂端元件(以一水平表示的極左端元件)是最低有效位元。在處理結束，特殊位址$Y_0$=0、$Y_1$=8、$Y_2$=9、$Y_3$=10包含輸出向量$y=(y_0,...,y_3)^T$的元件，其中$Y_0$是$y_0$的位址，$Y_1$及其他是$y_1$的位址。箭號係表示採用乘以1或-1符號的一位址內容的動作，並且將結果傳送而加入另一位址內容。一箭號係表示符號是1，而且雙箭號係表示符號是-1。此可根據在每個反複中的反複順序及位址(逐漸增加)順序完成。

範例3：$U_1$矩陣

下列步驟序列係描述在轉換矩陣的登錄屬於組$U_1$©{1,

五、發明說明（ 81 ）

-1,j,-j}情況中的GEM方法之一實施。GEM的一些情況之中的一者是時常在應用出現。資料是由一rxn $U_1$矩陣$A=(a_{ij}$；$0 \cdot i<r$, $0 \cdot j<n)$及一輸入複數向量$x=(x_0,\ldots,x_{n-1})$所組成。矩陣的欄是以$w_0,\ldots,w_{n-1}$表示。此一連串步驟可計算乘積$y=(y_0,\ldots,y_{r-1})^T=A \cdot x$。在每個特定反複上，每個位置包含一實數或複數，其是因向量x而定。配置的讀與寫記憶體包含從標示0到$4^{r-1}+r-2$的位址$4^{r-1}+r-1$。此範例係使用在先前範例中的定義組、及下列定義：

定義

1)藉由下列定義在組$U_1$與組{0,1,2,3}之間的一對應(及逆向對應)：

| | |
|---|---|
| 1'=0, | 0*=1 |
| (-1)'=1, | 1*=-1, |
| j'=2, | 2*=j, |
| (-j)'=3, | 3*=-j |

因此，對於r維矩陣的$U_1$向量而言，$u=(u_0,\ldots,u_{r-1})$係定義：

$$\Box(u)=\sum_{0\le j<r} 4^j u'_j.$$

2)對於每個r維矩陣的$U_1$向量而言，$u=(u_0,\ldots,u_{r-1})$係定義：

$$Sign(u)=u_{r-1}$$
$$g(u)=(u_0\cdot(u_{r-1})^{-1},\ldots,u_{r-1}\cdot(u_{r-1})^{-1}).$$

3)定義$Y_0=0$，及對於所有$1 \cdot m<r$而言：$Y_m=4^{r-1}+m-1$。這些將是輸出向量元件的位址。

五、發明說明（ 82 ）

4)對於所有k>0，j>1而言，映射$F_{k,i}$, $G_{k,j}$係依下列定義。讓l= l(k,j)、m= l(k+1,2j)、h = l(kj+1)-1。採用下列基於4表示的一整數v>0：$v=\Sigma_{0\le j<r-2}\cdot 4^j\cdot v_j$

而且定義如下：

$$Sign_{k,j}(v) = (v_{m-1})^* = (\lfloor (v \bmod 4^m)/4^{m-1} \rfloor)^*$$
$$F_{k,j}(v) = 4^m \cdot \lfloor v/4^m \rfloor$$

$$G_{k,l}(v) = \sum_{l\le j<m-1} 4^j \cdot ((v_j)^* \cdot ((v_{m-1})^*)^{-1}),$$

虛擬碼：

1.初始化：將零置於從0到$4^{r-1}+r-2$的每個記憶體

2.第一階段：for j是從0到n-1將$Sign(w_j)\cdot x_j$加到位址□(g($w_j$))

3.主要部分：

for k是從0到⌈log(r)⌉-1 do

for j是從1到$2^k$ do

if l(k,j+1)-1 > l(k,j) then：

1)將位址$Y_{l(k+1,2j)}$的值加到位址$Y_{l(k,j)}$的值

2)for p是從1到$4^{l(k,j+1)-l(k,j)-1}$-1 do

設定$u=4^{l(k,j)}\cdot p$及for來源位址u do

if $G_{k,j}(u)=u$ then

將(來源)位址u的值加到(目的地)位址$Y_{l(k+1,2j)}$的值

else if $F_{k,j}(u)=u$ then

將(來源)位址u的值加到(目的地)位址$Y_{l(k,j)}$的值

五、發明說明（ 83 ）

else if $G_{k,j}(u)=0$將乘以$Sign_{k,j}(u)$的（來源）位址u的值加到（目的地）位址$Y_{1(k,j)}$的值及亦將（位址u）值加到位址$F_{k,j}(u)$的值。

將乘以$Sign_{k,j}(u)$的（來源）位址u的值加到（目的地）位址$G_{k,j}(u)$及亦將（位址u）的值加到位址$F_{k,j}(u)$的值。

4.獲得輸出：

對於所有$0 \bullet i<r$而言，每個位址$Y_i$目前包含$y_i$的值。

說明

i)在上述實施$U_1$設定中的一基本步驟係表示從稱為來源的一記憶體位置讀取一複數，將它乘以1或-1或j或-j的一複數符號，並且將結果加到稱為目的地的另一記憶體位置中放置的一複數。

ii)複數可由下列各項形成。對於$0 \bullet k \bullet \lceil \log(r) \rceil -1$及對於$1 \bullet j \bullet 2^k$而言，讓：

$$w_{k,j} = 4^{l(k,j+1)-l(k,j)-1}$$

$$w_k=\sum_{1 \bullet j \bullet 2^k} w_{k,j} = \sum_{1 \bullet j \bullet 2^k} 4^{l(k,j+1)-l(k,j)-1}$$

對於$k = \lceil \log(r) \rceil$及對於$1 \bullet j \bullet 2^k$而言，讓：

$$w_{k,j}=\lfloor 4^{l(k,j+1)-l(k,j)-1} \rfloor$$

$$w_k=\sum_{1 \bullet j \bullet 2^k} w_{k,j} = r.$$

ii)基於上述虛擬碼的一裝置需要下列基本步驟實施$A \bullet x$計算。

(1)對於第一階段而言，n個基本步驟是需要的。

五、發明說明（ 84 ）

(2)對於所有0•k•⌈log(r)⌉-1及1•j•$2^k$而言，下列整個

$$2•w_{k,j}-w_{k+1,2j} - w_{k+1,2j-1} + 1$$

基本步驟可在步驟(k,j)完成。

(3)因此，對於所有0•k•⌈log(r)⌉-1而言，主要部分的k重複需要下列基本步驟：

$$2^k + 2•w_k-w_{k+1}$$

(4)如此，A•x的上述計算的主要部分需要下列基本步驟：

$$C^{U_i}_r \text{©} + 2w_0 + w_1 + w_2 + \ldots\ldots + w_{\lceil \log(r) \rceil - 1} + 2^{\lceil \log(r) \rceil} - 1 - r.$$

一數值只因r而定。

(5)如此，整個計算A•x需要下列整個

$$C^{U_1}_{n,r} \text{©} n + C^{U_1}_r$$

基本步驟。

範例4：Topiltz U矩陣

下列一連串步驟係描述使用U係數之本發明的Toplitz矩陣觀點的實施。資料是由一U序列$t_0,\ldots\ldots,t_{n-1}$及一輸入實數、或複數向量$x=(x_0,\ldots\ldots,x_{n-r-2})$所組成。從U序列，rx(n+r-1)Toplitz矩陣A©($a_{ij}$©$t_{i-j}$：0•i<r, 0•j•n+r-2)可被形成，其中$t_k$©0對於整個而言，k<0或k-n。這些步驟可計算乘積$y=(y_0,\ldots\ldots,y_{n-1})=A•x$。只有第一階段是不同於U矩陣範例，因此，需要只在此階段使用。在實施的先前範例列中列出的所有定義在此可適用。

五、發明說明（ 85 ）

虛擬碼

1.初始化：將零置於從0到$2^{r-1}+r-2$的每個記憶體

2.第一階段：

1)for j是從0到r-2將$(1/2)\cdot l_{n-r+j+1}\cdot x_j$加到位址

$\pi(h(t_j,t_{j-1},\ldots\ldots t_1,t_0,\ t_{n-1},t_{n-2},\ldots\ldots,t_{n-r+j+1}))$

及亦將$-(1/2)\cdot l_{n-r+j+1}\cdot x_j$加到位址

$\pi(h(t_j,t_{j-1},\ldots\ldots t_1,t_0,\ -t_{n-1},-t_{n-2},\ldots\ldots,-t_{n-r+j+1}))$

2)far j是從r-1到n-1將$t_{j-r+1}\cdot x_j$加到位址

$\pi(h(t_j,t_{j-1},\ldots\ldots,t_{j-r+1}))$

3)for j是從n到n+r-2將$(1/2)\cdot t_{j-r+1}x_j$加到位址

$\pi(h(t_{j-n},t_{j-n-1},\ldots\ldots t_1,t_0,\ t_{n-1},t_{n-2},\ldots\ldots,t_{j-r+1}))$

亦將$(1/2)\cdot t_{j-r+1}x_j$到位址

$\pi(h(t_{j-n},t_{j-n-1},\ldots\ldots,t_1,-t_0,\ -t_{n-1},-t_{n-2},\ldots\ldots,-t_{j-r+1}))$

3.主要部分：演算法目前可在實施一演算法的主要部分的U矩陣中處理，而且輸出是儲存的在相同位址。

說明

i)上述Toplitz實施的基本步驟是與U實施相同，即是，從稱為來源的一記憶體位置讀一數值，將它乘以1或-1符號，然後將結果加到稱為目的地的一另一記憶體放置的一數值。

五、發明說明（ 86 ）

ii)基於上述虛擬碼的一裝置需要下列基本步驟來實施A•x計算。

(1)對於第一階段而言：

$$2\bullet(r-1)+n-r+1+2\bullet(r-1)=n+3r-3$$

基本步驟是需要的。

(2)A•x的Toplitz計算的主要部分可藉由具r列的U碼的一主要部分實施達成。如此，它需要$C^{U}_{r}$基本步驟。

(3)如此，A•x的整個Toplitz計算需要下列整個

$$C^{T}_{n,r} \copyright\ n + 3r-3 + C^{U}_{r}$$

基本步驟。

圖解3：Toplitz矩陣

圖3係描述將輸入資料傳送給由一裝置所採用的適當記憶體位置，以執行上述虛擬碼的初始部分，而且Toplitz矩陣是由4列組成。除了此初始部分之外，每個其他觀點是與U矩陣實施與裝置相同，而且描述可在此適用。在此，一箭號係表示符號是1，而且雙箭號係表示符號是-1。

範例5：Topiltz $U_l$矩陣

本發明的下列較佳具體實施例是具$U_l$係數之本發明的Toplitz矩陣觀點實施。資料是由一$U_l$序列$t_0,.....,t_{n-1}$及一輸入複數向量$x=(x_0,.....,x_{n+r-2})$所組成。從$U_l$序列，一$rx(n+r-1)$Toplitz矩陣$A\copyright(a_{i,j}\copyright t_{i-j}：0\bullet i<r, 0\bullet j\bullet n+r-2)$可被形成，其中$t_k\copyright 0$於所有情況是$k<0$或$k=n$。下面列出的一連串步驟可計算乘積$y=(y_0,.....,y_{n-1})=A\bullet x$。只有第一階段是不同於$U_l$矩陣

五、發明說明（ 87 ）

範例，因此只有此階段需要描述。在先前範例列出的所有定義在此可適用。

虛擬碼：

1.初始化：將零置於從0到$4^{r-1}+r-2$的每個記憶體

2.第一階段：

1)for j是從0到r-2將$(1/2)\bullet t_{n-r+j+1}\bullet x_j$加到位址

□$(g(t_j,t_{j-1},\ldots\ldots\ldots t_1,t_0,\ t_{n-1},t_{n-2},\ldots\ldots\ldots,t_{n-r+j+1}))$

亦將$-(1/2)\bullet t_{n-r+j+1}\bullet x_j$加到位址

□$(g(t_j,t_{j-1},\ldots\ldots\ldots t_1,t_0,-t_{n-1},-t_{n-2},\ldots\ldots\ldots\ldots,-t_{n-r+j+1}))$

2)for j是從r-1到n-1將$t_{j-r+1}\bullet x_j$加到位址

□$(g(t_j,t_{j-1},\ldots\ldots\ldots,\ t_{j-r+1}))$

3)for j是從n到n+r-2將$(1/2)\bullet t_{j-r+1}x_j$加到位址

□$(g(t_{j-n},t_{j-n-1},\ldots\ldots t_1,t_0,\ t_{n-1},t_{n-2},\ldots\ldots,t_{j-r+1}))$

亦將$(1/2)\bullet t_{j-r+1}x_j$加到位址

□$(g(t_{j-n},t_{j-n-1},\ldots\ldots,t_1,-t_0,\ -t_{n-1},-t_{n-2},\ldots\ldots,-t_{j-r+1}))$

亦將$-(1/2)\bullet t_j x_j$加到位址

□$(g(t_{j-n},t_{j-n-1},\ldots\ldots,t_1,-t_0,\ -t_n,-t_{n-1},\ldots\ldots,-t_{j-r+2}))$

3.主要部分：可在$U_1$矩陣演算法的主要部分中處理，而

五、發明說明（ 88 ）

且資料能以相同方式接收。

說明

i)上述Toplitz實施的一基本步驟是與$U_1$實施相同，即是，從稱為來源的一記憶體位置讀取一複數，將它乘以1或-1或j或-j的複數符號，然後將結果加到稱為目的地的另一記憶體位置中放置的一複數。

ii)基於上述虛擬碼的一裝置需要下列基本步驟實施A•x計算。

(1)對於第一階段而言

$$2\bullet(r-1)+n-r+1+2\bullet(r-1)=n+3r-3$$

基本步驟需要。

(2)A•x的Toplitz計算的主要部分是經由具r列的$U_1$碼的主要部分實施達成。如此，它需要$C^{U_1}{}_r$基本步驟。

(3)如此，A•x的整個Toplitz計算需要下列：

$$C^{T_1}{}_{n,r} \odot n+3r-3 \quad +C^{U_1}{}_r$$

整個基本步驟。

範例6：實數矩陣、二進位0-1表示。

本發明的下列較佳具體實施例是使用矩陣登錄的二進位0-1轉換之本發明的實數矩陣觀點實施。資料是由具負登錄的一rxn實數矩陣$A=(a_{ij}: 0\bullet i<r, 0\bullet j<n)$及一輸入實數或複數向量$x=(x_0,.....,x_{n-1})$所組成。演算法可計算$y=(y_0,.....y_{n-1})=A\bullet x$。假設，對於所有$0\bullet i<r$, $0\bullet j<n$而言，矩陣的登錄是由下列0-1二進位表示提供：

五、發明說明（ 89 ）

$$a_{ij} = \square_{-m_2 \le k \le m_1} t_{ijk} \cdot 2^k$$

其中：對於所有$-m_2 \cdot k \cdot m_1$而言，$t_{ijk} \in \{0,1\}$

只有第一階段是不同於0-1矩陣實施，因此，只有此階段需要描述。在0-1矩陣實施部分中列出的所有定義可在此適用。此外，對於所有$0 \cdot j < n$, $-m_2 \cdot k \cdot m_1$而言，可定義下列r維矩陣{0,1}向量：

$$v_{jk} = (t_{1jk}, t_{2jk}, \ldots\ldots\ldots\ldots, t_{rjk})$$

1.初始化：將零置於從0到$2^r-1$的每個記憶體

2.第一階段：

for一計數j是從0到n-1 do

for k是從$-m_2$到$m_1$ do

將$2^k \cdot x_j$加到位址$\square(v_{jk})$的值

3.主要部分：它可在具r列的{0,1}矩陣實施演算法的主要部分中處理，而且在計算處理結束，輸出可儲存在相同位址。

說明

i)上述實施的基本步驟是與0-1二進位實施相同，即是，從稱為來源的一記憶體位置讀取一數值，並且將它加到稱為目的地的另一記憶體位置中放置的一數值。

ii)基於上述虛擬碼的一裝置需要下列基本步驟，以實施$A \cdot x$計算。

(1)對於第一階段而言

$$(m_1 + m_2 + 1) \cdot n$$

五、發明說明（ 90 ）

基本步驟是需要的。

(2)虛擬碼的主要部分可藉由具r列的上述實施0-1二進位碼的主要部分達成。如此，它需要$C^{0,1}_{r}$個基本步驟。

(3)如此，A•x的整個計算需要下列整個：

$$C^{Re\text{-}0\text{-}1} \odot (m_1 + m_2 + 1)\cdot n + C^{0,1}_{r}$$

基本步驟。

範例7：實數矩陣、二進位U表示法

本發明的下列較佳具體實施例是具矩陣登錄的二進位U表示法之本發明的實數矩陣觀點實施。資料是由一rxn實數矩陣$A=(a_{ij}：0•i<r, 0•i<r, 0•j<n)$及一輸入實數或複數向量$x=(x_0,.....,x_{n-1})$所組成。演算法可計算$y=(y_0,.....,y_{n-1})=A•x$。假設，對於所有0•i<r, 0•j<n而言，矩陣的登錄可藉由下列U表示法提供，

$$a_{ij} = \square_{-m_2-1 \le k \le m_1-1} t_{ijk}•2^k + t_{ijm_1}•(2^{m_1} - 2^{-m_2-1})$$

其中：對於$-m_2-1•k•m_1$而言，$t_{ijk}\in U$

此表示法已在本發明的實數矩陣觀點中提到。只有第一階段是不同於U矩陣實施。因此，只有此階段需要描述。在U矩陣實施部分中列出的所有定義在此可適用。此外，對於所有0•j<n, $-m_2-1•k•m_1$而言，可定義下列r維矩陣U向量：

$$u_{jk}=(t_{1jk}, t_{2jk},.....,t_{rjk})$$

1.初始化：將零置於從0到$2^r+r-1$的每個位址

2.第一階段：

五、發明說明（ 91 ）

for j是從0到n-1 do

for k是從$-m_2-1$到$m_1-1$ do

將$2^k \cdot Sign(u_{jk}) \cdot x_j$加到位址$\pi(h(u_{jk}))$

for $k=m_1$ do

將$(2^{m1}-2^{-m2-1}) \cdot Sign(u_{jk}) \cdot x_j$加到位址$\pi(h(u_{jk}))$

3.主要部分：它可在U矩陣實施演算法的主要部分r列中處理，而且在計算處理結束，輸出可儲存的相同的位址。

說明

i)上述實施的一基本步驟是與上述U實施相同，即是，從稱為來源的一記憶體位置讀取一數值，及將它乘以1或-1的一符號，然後將此結果加到位於稱為目的地另一記憶體位置的一數值。

ii)根據上述碼的一裝置將需要下列基本步驟數量，以實施$A \cdot x$的計算。

(1)對於第一階段

$(m_1 + m_2 + 2) \cdot n$而言，

基本步驟是需要的。

(2)主要的碼部分是由上述實施具r列U碼的主要部分達成。如此，它需要$C^{U}_{r}$個基本步驟。

(3)$A \cdot x$的整個計算如此需要下列整個步驟：

$C^{Rz-U}$ ©$(m_1+m_2+1) \cdot n+C^{u}_{r}$

之基本步驟。

圖4係根據本發明的一較佳具體實施例而使用一減少加

五、發明說明（ 92 ）

算數量執行一線性轉換的裝置方塊圖。裝置500是由下列組成：一乘法器10，其具有2個輸入及1個輸出；一多工器(MUX)9，用以選取輸入，以便將該等輸入傳送給輸出；一加法器11，其具有2個輸入及1個輸出；及一雙埠隨機存取記憶體(DPRAM)13，其具有2個位址匯流排線"add_a"和"add_b"、及2個輸出"data_a"和"data_b"。MUX動作會受到一位址產生器501的控制，而可存取DPRAM 13的記憶體位址。位址產生器動作可經由一計數器3控制。

乘法器10的輸出係連接到MUX 9的一輸入"C"。MUX 9的輸出係連接到加法器11的一輸入"A"。加法器11的輸出係連接到DPRAM 13的輸入。DPRAM 13的一輸出"data_a"係連接到加法器11的輸入"B"。DPRAM 13的另一輸出"data_b"係連接到一乘法器12的輸入"E"。乘法器12的另一輸入"F"係連接到位址產生器501的輸出"符號"。乘法器12的輸出係連接到MUX 9的另一輸入"D"。計數器3係連接到位址產生器501的2個輸入。產生器501的輸出"H"係連接到乘法器12的"符號"輸入。產生器501的輸出"G"係連接到MUX 9的控制輸入"S"。產生器501的輸出"J"係連接到DPRAM 13的第一位址輸入"add_a"。產生器501的另一輸出"I"係連接到DPRAM 13的第二位址信號輸入"add_b"。轉換矩陣可儲存在一選擇性RAM/ROM 1，其可將一連串碼(矩陣的一列位元)$D_0, D_1, \ldots, Ds$提供給位址產生器501與乘法器10，其中最高有效位元是$D_0$。輸入向量可儲存在另一選擇性RAM/ROM 2，其可將輸入信號的取樣提供給乘法器10。或者，如果輸

五、發明說明（ 93 ）

入向量的元件及轉換矩陣的對應元件同步提供給裝置500，儲存記憶體1和2便可移除。例如，這些元件可由一ADC或一序列產生器提供。裝置的所有元件500可經由"時脈輸入"輸入而受到一共同時脈的控制。

藉由使用代表相同不同碼$D_0,D_1,....,D_s$的相同U矩陣(包含n列)的相同輸入組$[x_1,x_2,.....,x_n]$，裝置的操作是在下面說明。裝置的操作500可分成2個階段。標示"階段1"的第一階段，在其輸入資料(亦即，取樣$[x_1,x_2,.....,x_n]$)接收與每個元件及轉換矩陣的對應元件計算及儲存在DPRAM 13期間；及標示"階段2"的第二階段，在接收資料處理而阻斷進入加法器9的進一步輸入資料期間。計數器3可計算運算的累積次數，而且計數(計數器的輸出)可用來兩階段之間的區別。"階段1"的運算次數是輸入向量的長度n。超過此數的運算是與"階段2"有關。

根據本發明的一較佳具體實施例，位址產生器501包含一比較器4，該比較器係連結到計數器3的輸出。比較器可讀取計數器3的目前輸出，將它與輸入向量的長度n比較，並且提供一對應信號，以表示目前運算是否在"階段1"或"階段2"實施。此信號可用來控制MUX 9的輸入"S"，如此可在輸入"C"與"D"之間切換。

位址產生器501亦包含可儲存經程式化值的一非同步記憶體5(例如，一ROM)，並且可當作一查表(LUT)使用，其可經由用以處理內容的輸入位址"add_a"和"add_b"而決定DPRAM 13的位址。LUT的大小是(C-n)x(1+2r)，而且它的

五、發明說明（ 94 ）

內容包含3個欄位，一"來源"欄位、一"符號"欄位及一"目的地"欄位。對於每個運算(亦即，計數器3計算的每個時脈週期)而言，有3個對應來源符號及目的地值。來源欄位包含與轉換矩陣的分離(分開)每個欄位有關的資訊、及與特殊欄的兩部分正常化有關的一指示。來源欄位可決定DPRAM 13的輸入"add_b"值。符號欄位係表示在每個分開欄或子欄中的較低元件。目的地欄位可根據對應位址的內容的處理選取而決定DPRAM 13的輸入"add_a"值。

位址產生器亦包含一組s(s=r-1)反相器$6_1,6_2,....,6s$，其每個包含分別連接到一連串位元$D_1,...,Ds$的一輸入。每個反相器的輸出是從一組s個多工器$7_1,7_2,...,7s$連接到一對應MUX的一輸入。一連串位元$D_1,....,Ds$亦從一組s個多工器$7_1,7_2,...,7s$提供給一對應MUX的另一輸入。來自多工器組$7_1,7_2,...,7s$的每個MUX輸出可由最高有效位元$D_0$值控制，如此，可將未改變組$D_1,...,Ds$或相反組(亦即，$D'_1,...,D's$)傳送給DPRAM 13的輸入"address_a"。未改變組$D_1,.....,Ds$(或$D'_1,...,Ds$)可輸入s個多工器的一對應組$8_1,8_2,.....,8s$的一輸入；及一額外多工器8，並且將來自LUT的"add_a"的MSB(第r位元)提供給一額外多工器8。比較器4的輸出允許"add_a"的選擇，以便從多工器組$7_1,7_2,...,7s$或從LUT到達。輸入"add_a"(亦即，目的地)可控制DPRAM 13的輸出"data_a"，該輸出"data_a"可提供給加法器11的輸入"B"。從LUT(亦即，來源)取得的輸入"add_b"可控制DPRAM 13的第二輸出"data_b"，該輸出"data_b"可提供給乘法器12的輸入

五、發明說明（ 95 ）

"E"，將"data_b"的每個值乘以從LUT擷取的一對應"符號"值，並且將乘積提供給乘法器12的輸入"D"。DPRAM 13的"寫"操作可同步，亦即，每單元的內容可根據時脈率(例如，當時脈信號上升時)而重寫。另一方面，DPRAM 13的"讀"操作是非同步，亦即，當位址輸入改變時，每個輸出便改變，而不管時脈。

階段1的操作：

在此階段，計數器3可在階段1執行期間計算第一符號時間(n時脈週期)。在階段1期間，MUX 9允許來自它輸入"C"的資料流，以便流入加法器11的輸入"A"，而阻斷輸入"D"。輸入符號[$x_1$,$x_2$,.....,xn]可提供給乘法器10的一輸入。碼的最高有效位元$D_0$可提供給乘法器10的另一輸入。在此階段，由輸入"add_a"決定的目的地(輸出"data_a")可從DPRAM 13擷取，並且加入乘以最高有效位元$D_0$的輸入向量元件[$x_1$,$x_2$,.....,xn]。計數器3，計算目前符號時間的計數器3可在目前符號時間結束時將一指示提供給位址產生器比較器4及ROM 5，而且比較器4可將MUX 9的輸入選擇從輸入"C"改變成另一輸入"D"。同樣地，比較器4可驅動多工器$8_1$,$8_2$,.....,8s，以便從LUT選取資料，而不是從RAM 1。

階段2的操作：

在此階段，計數器3可在階段2執行期間開始計算下一符號時間。階段2期間，MUX 9允許資料從輸入"D"流入它的輸出及加法器11的輸入"A"，而阻斷輸入"C"。在此階段，於每個時脈週期，在DPRAM 13的一選取位址可經由表示來

五、發明說明（ 96 ）

源的"add_b"而存取。來源資料可從DPRAM 13的第二輸出"data_b"提供給乘法器12的輸入"E"，而乘以從LUT擷取的對應"符號"值。此乘積可提供給MUX 9的輸入"D"，藉使在加法器11的輸入"A"上出現。同時，目的地值的內容可在加法器11的輸入"B"出現。2個值可經由加法器11相加，而且結果可儲存在DPRAM 13(其對應先前目的地值)的相同位址。在此加總處理結束，對應r轉換點($y_i$'S)可儲存在位址#0及#$2^{r-1}$至($2^{r-1}$+ r- 2)的DPRAM 13。此處理因此可持續，直到所有轉換點($y_i$'S)已計算及儲存在不同DPRAM 13的不同位址之後，時脈提供階段2結束(根據一預定計數)的一指示為止。此時，目前符號時間亦已結束，而且比較器4可將MUX 9的輸入選擇從輸入"D"改變回到另一輸入"C"，，並且驅動多工器$8_1$,$8_2$,....,8s，以便從RAM 1選取資料，而不是從LUT，藉此驅動裝置500，以便在階段1重新操作。

用來實施上述方法的一裝置具體實施例可參考圖5。圖5係描述一rxn U矩陣與一n維矩陣向量X乘積的一實施。該矩陣的表示包含0或1，其中0是對應1，而且1是對應-1。

在此範例中，向量是實數，而且v位元是專屬於每個元件。在隨後的結構中，矩陣A是儲存在元件1(RAM或ROM)，而且向量X是儲存在元件2(RAM或ROM)。矩陣與向量可從例如記憶體裝置或同步來源的任何內部或外部裝置產生，例如ADC或一些序列產生器等。

控制模組1、2、3、13的一時脈能與整個系統同步。模組3是從0到C計算的一計數器，其中C的定義如下示：

五、發明說明（ 97 ）

$$C = n + 2u_0 + u_1 + u_2 + \ldots\ldots + u_{\lceil \log(r) \rceil - 1} - r.$$

$$u_k = \sum_{1 \le j \le 2^k} 2^{h(k,j)-l(k,j)}.$$

$$l(k,j) = \lceil (j-1) \cdot (r/2^k) \rceil$$

$$h(k,j) = \lceil j \cdot (r/2^k) \rceil - 1$$

在階段1，來自元件1&2的輸入信號可插入符合來自元件3位址的元件13，其中每個位址包含$\log_2 n$個有效位元。

此外，計數器可當作(C-n)x(1+r+r)大小的一非同步ROM元件5的一位址產生器使用。計數器的所有位元可進入一比較器的元件4，以檢查目前計數是否大於n。元件5包含三個欄位：符號、來源、目的地。在ROM的線條順序應該是在n個週期之後，階段2的第一操作可由計數器引起。

對於語法目的而言，我們將可定義s=r-1。資料匯流排可分別元件1位元D1-Ds與元件元件6_1-6_s(反相器)出發。元件71-7s可根據D0的狀態而在D1-Ds與61-6s的輸出之間選取。

(D0=0=>7_1-7_s=D1-Ds,D0=1=>71-7s=61-6s).

元件8_1-8_s可根據元件4(階段1_2n)的比較器輸出狀態而在元件5(add_a=目地)的7_1-7_s與$\log_2(n)$LSB的輸出之間選取。階段1-2n=0=>8_1-8_s=add_a，階段1_2n=1=>8_1-8_s-7_1-7_s。

元件8_r可在元件5的"0"與r位元(add_a MSB)之間選取。階段1_2n=0=>8_r = add_a MSB，階段1_2n =1=>8_r=0。來自8_1-8_r的輸出可當作元件13的第一位址匯流排使用，其

五、發明說明（ 98 ）

是在$(2^{r-1}+r-1)x(V+\log_2 n)$大小上的一雙埠RAM。此位址是定義從元件13輸出的data_a匯流排，而且亦是在輸入元件13之data_in匯流排的目的地。

Add_b(其是元件5(來源領域)的輸出)是元件13(此位址可控制從元件13輸出的data_b匯流排)的第二位址匯流排。

符號是從元件13輸出的單一位元，而且在元件12(一乘法器)乘以data_b。

有關元件13的一輸入data_in匯流排是從元件11到達，其中該元件11是一加法器，其以可將data_a與元件9的輸出加總。data_in能以一同時方式到add_a目的地，而data_a和data_b的讀操作是非同步。

在元件13動作之前，它可藉由插入零而開始。元件9可在元件10與元件12的輸出之間選取。元件10可將來自元件2的資料匯流排與元件1的最低有效位元相乘。在C週期之後，結果向量可儲存在元件13的位址0及位址$2^{r-1}$到$(2^{r-1}+r-2)$。

本發明的許多變化與修改對於在技藝中熟諳此技者是顯然的。因此，本發明能以其他特殊形式具體實施，而不致於違背其精神或必要特性。詳細的具體實施例在各方面可考慮，而並未只局限於在此描述及本發明的範圍。因此，本發明的範圍如附錄申請專利所述，而不是先前的描述。所有變更是在附錄申請專利的意義與範圍內。

A5
B5

四、中文發明摘要（發明之名稱：用以有效執行線性轉換之方法及裝置）

一種用以執行線性轉換之改良方法及裝置，且可減少算術運算次數、簡化電路、及低消耗功率。該方法可執行在CDMA應用中出現的複數U轉換。該裝置允許在不同展開因素中同時分析一輸入信號。

英文發明摘要（發明之名稱：METHOD AND APPARATUS FOR EFFECTIVELY PERFORMING LINEAR TRANSFORMATIONS）

An improved method and apparatus for performing linear transformations, with reduced number of arithmetic operations, simplified circuitry, and low power consumption. The method performs complex U-transformation which appear in CDMA applications. The apparatus allows for simultaneous analysis of an incoming signal in different spreading factors.

## 六、申請專利範圍

1. 一種用以提高藉由上述實數或複數或一有限欄位的至少一n維矩陣輸入向量之一r×n矩陣所表示一線性轉換性能效率之方法，其包含：

   將在每個省略列與每個選取列之間的一比率儲存在記憶體；

   省略該矩陣的零欄及該輸入向量的對應數量元件；

   使各欄或該矩陣正常化；

   從正常化矩陣中的相等欄群產生一修改的向量；

   產生一修改的矩陣；及

   獲得該輸出向量。

2. 如申請專利範圍第1項之方法，其進一步包含藉著使每個子矩陣乘積的輸出向量相同，而將轉換矩陣分成數個子矩陣及獲得該輸出向量。

3. 如申請專利範圍第2項之方法，其中該修改的矩陣包含該轉換矩陣的一部分列。

4. 如申請專利範圍第3項之方法，其進一步包含將輸入向量分成數個子向量，以致於每個子向量可對應一子矩陣，而且其中該輸出向量可藉著增加由每個子矩陣乘積產生的該等輸出向量而獲得。

5. 如申請專利範圍第1項之方法，其進一步包含將一修改的矩陣分成數個子矩陣，其中一輸出向量可藉由相對子向量而透過增加每個子矩陣乘積產生的該等輸出向量獲得。

6. 如申請專利範圍第1項之方法，其進一步包含藉著將欄乘上一先導元件的導數而使該矩陣的各欄正常化。

六、申請專利範圍

7. 如申請專利範圍第1項之方法，其中該輸出向量是該矩陣與該輸入向量的一乘積。

8. 如申請專利範圍第1項之方法，其進一步包含識別該正常化矩陣的相等欄群及將唯一位置加入每個識別群。

9. 一種用以執行一線性轉換之裝置，其包含：

第一及第二輸入，其可接收輸入資料及預定資料；

轉換電路，其可在該等輸入資料及預定資料動作；

控制及位址產生電路，其是連接到一第一記憶體，以產生對應位址，用以存取該等記憶體單元的對應位址，及用以控制在資料經由該第一輸入接收的一資料接收模式及經由該第一輸入的輸入資料到達阻滯的一資料處理模式之間的選擇；及

計數器電路，用以控制該裝置的運算時序。

10. 如申請專利範圍第9項之裝置，其中該轉換電路可將該輸入資料的每個元素乘以該轉換資料的一對應元件。

11. 如申請專利範圍第10項之裝置，其中該轉換電路包含一記憶體，用以儲存該乘法的結果。

12. 如申請專利範圍第9項之裝置，其中該轉換電路包含一加法與累積電路。

13. 如申請專利範圍第9項之裝置，其進一步包含一多工器電路，用以在該資料接收模式與該資料處理模式之間選擇。

14. 如申請專利範圍第9項之裝置，其中該控制與位址產生電路包含：

一第二記憶體，用以儲存預先程式化處理與控制資料

六、申請專利範圍

及；

一比較器電路，用以在該資料接收模式與該資料處理模式之間切換。

15. 如申請專利範圍第14項之裝置，其中該控制與位址產生電路進一步包含：

一第一組多工器，其每個多工器具有至少一直接輸入，用以接收轉換資料；及另一輸入，其中該轉換資料可經由一對應的反相器提供，該第一組多工器可被控制，而可經由該轉換資料提供之一預定值而將轉換資料或反轉的轉換資料轉移；

一第二組多工器，每個多工器具有至少一輸入，該輸入是連接到從該第一組多工器選取的一對應多工器輸出，和另一輸入，該第二組可被該比較器電路控制，以便藉著將來自該第一組的每個多工器輸出傳輸給來自該第二組的對應多工器輸出而將一第一位址提供給第一記憶體，及藉著傳輸在該第二記憶體中儲存的資料而將至少一部分第二位址提供給該第一記憶體；及

一多工器，其可在該資料處理模式與該第二組多工器結合操作，該多工器具有一未連接輸入及一輸入，其中該輸入是連接到該第二記憶體，而且該多工器可被該比較器電路控制，藉此提供該第二位址的其餘部分。

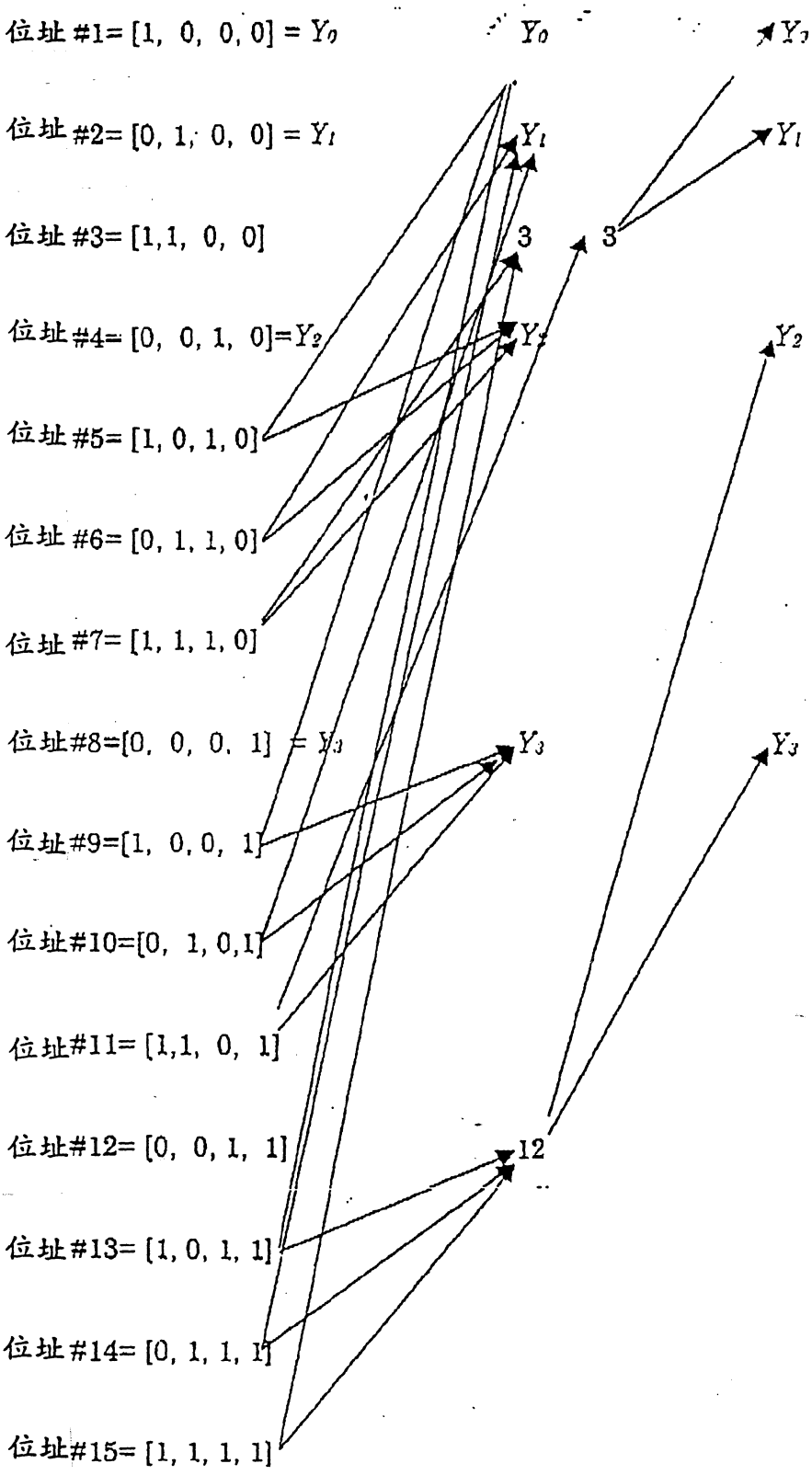
位址 #1= [1, 0, 0, 0] = Y0
位址 #2= [0, 1, 0, 0] = Y1
位址 #3= [1,1, 0, 0]
位址 #4= [0, 0, 1, 0]=Y2
位址 #5= [1, 0, 1, 0]
位址 #6= [0, 1, 1, 0]
位址 #7= [1, 1, 1, 0]
位址#8=[0, 0, 0, 1] = Y3
位址#9=[1, 0, 0, 1]
位址#10=[0, 1, 0,1]
位址#11= [1,1, 0, 1]
位址#12= [0, 0, 1, 1]
位址#13= [1, 0, 1, 1]
位址#14= [0, 1, 1, 1]
位址#15= [1, 1, 1, 1]
Y0
Y1
3
Y2
Y3
12
3
Y0
Y1
Y2
Y3

圖 1

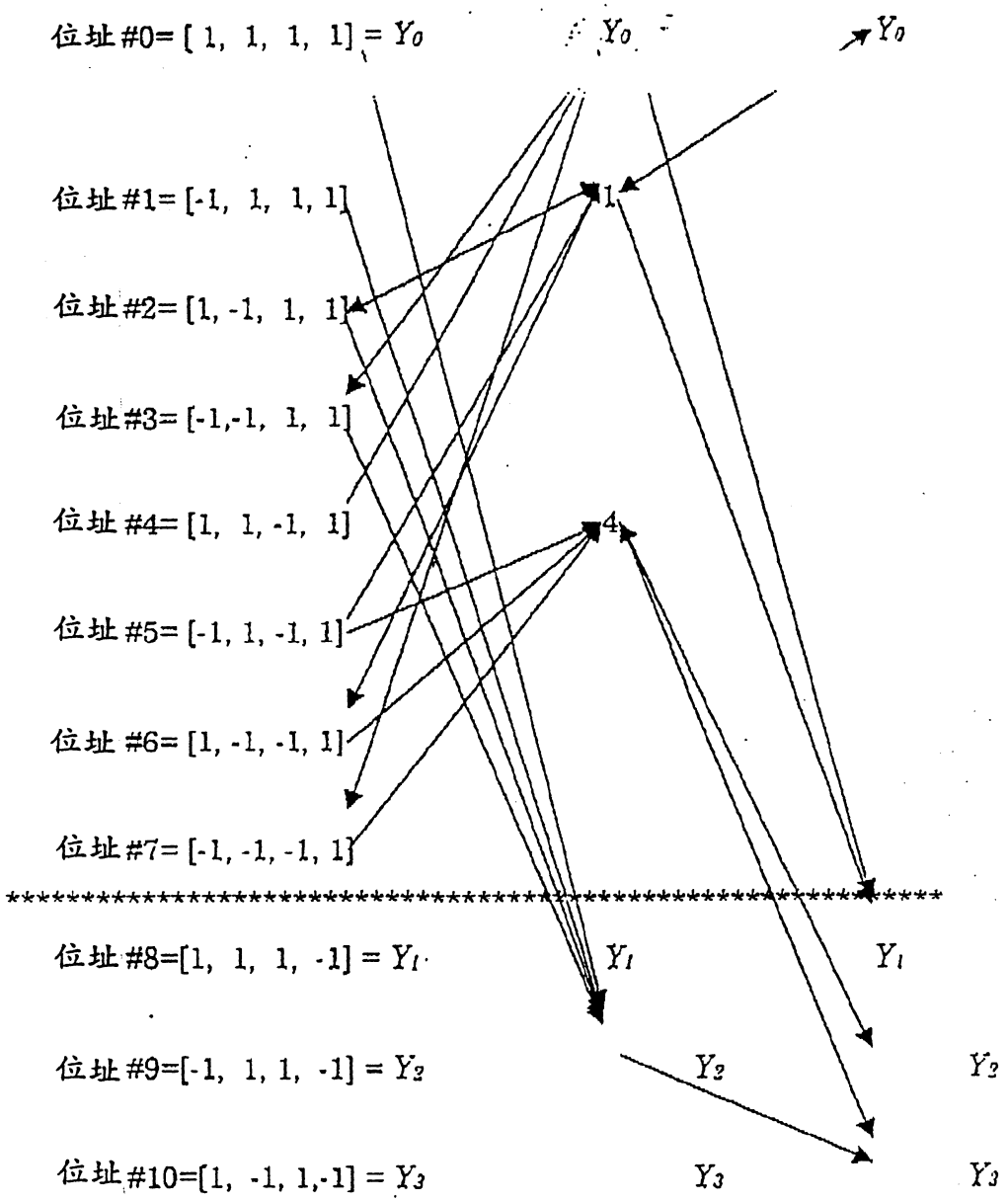


圖 2

Example 3: $U_1$ - Matrix

下列連續步驟係描述在屬於組U轉換矩陣登錄情況的一GEM的方法實施

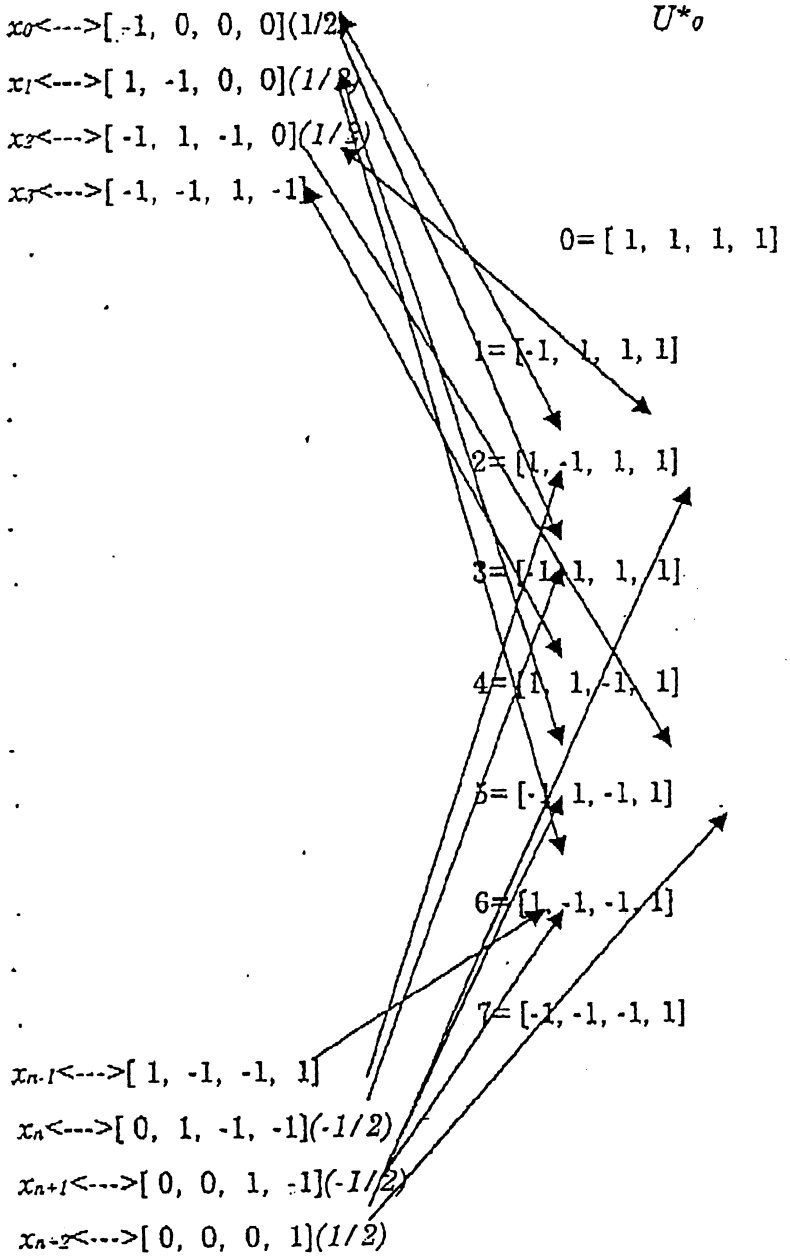
$x_0$<---->[ -1, 0, 0, 0](1/2)
$x_1$<---->[ 1, -1, 0, 0](1/2)
$x_2$<---->[ -1, 1, -1, 0](1/2)
$x_3$<---->[ -1, -1, 1, -1]
$U^*_0$
0= [ 1, 1, 1, 1]
1= [-1, 1, 1, 1]
2= [1, -1, 1, 1]
3= [-1, 1, 1, 1]
4= [1, 1, -1, 1]
5= [-1, 1, -1, 1]
6= [1, -1, -1, 1]
7= [-1, -1, -1, 1]
$x_{n-1}$<---->[ 1, -1, -1, 1]
$x_n$<---->[ 0, 1, -1, -1](-1/2)
$x_{n+1}$<---->[ 0, 0, 1, -1](-1/2)
$x_{n+2}$<---->[ 0, 0, 0, 1](1/2)

圖 3

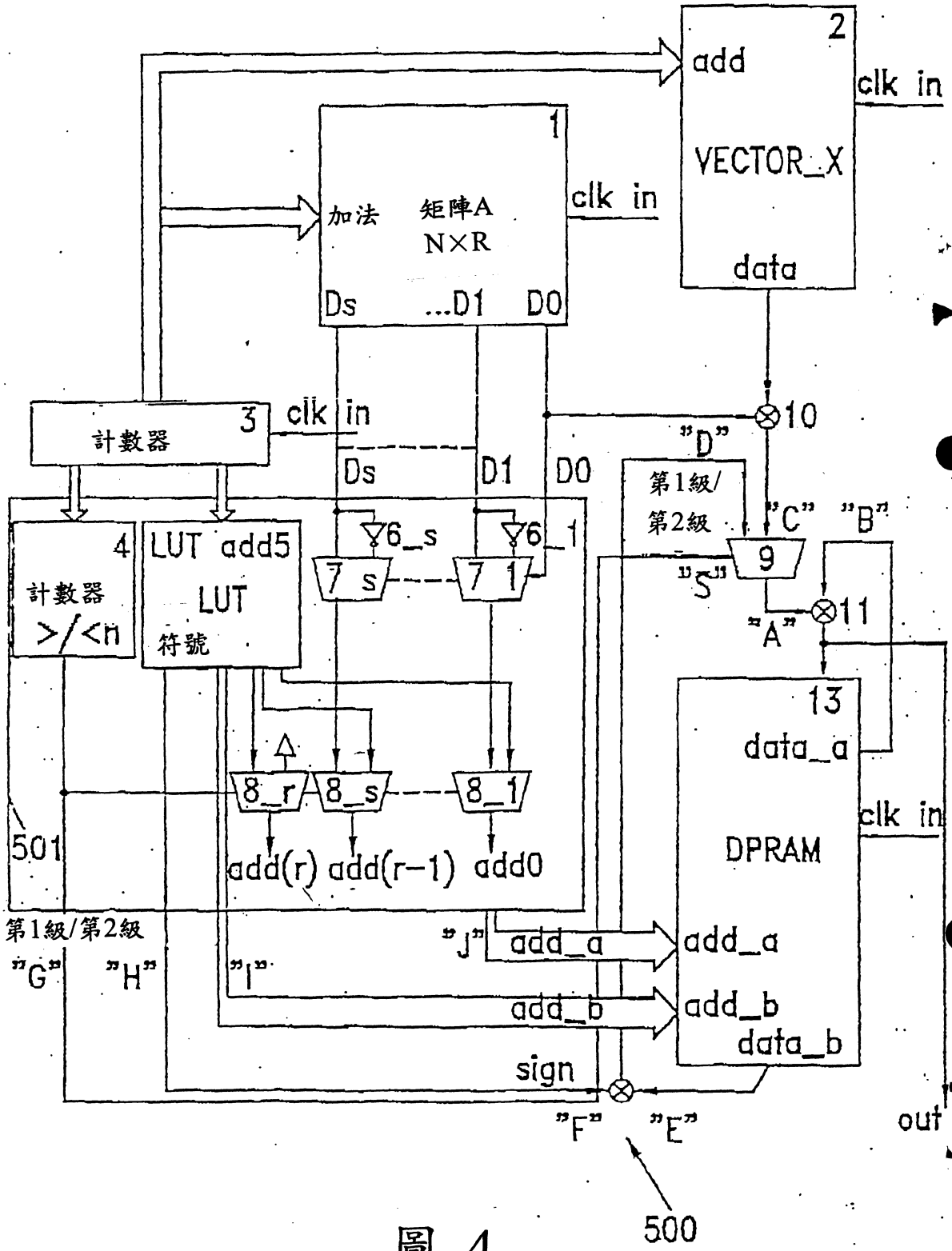
2
add
clk in
VECTOR_X
data
1
加法 矩陣A
N×R
clk in
Ds ...D1 D0
計數器 3
clk in
Ds D1 D0
10
"D"
第1級/
第2級
"C" "B"
9
"S"
11
"A"
4
計數器
>/<n
LUT add5
LUT
符號
6_s
6_1
7_s
7_1
8_r 8_s 8_1
add(r) add(r-1) add0
501
第1級/第2級
"G" "H"
"I"
"J"
add_a
add_b
13
data_a
clk in
DPRAM
add_a
add_b
data_b
sign
"F" "E"
out
500

圖 4

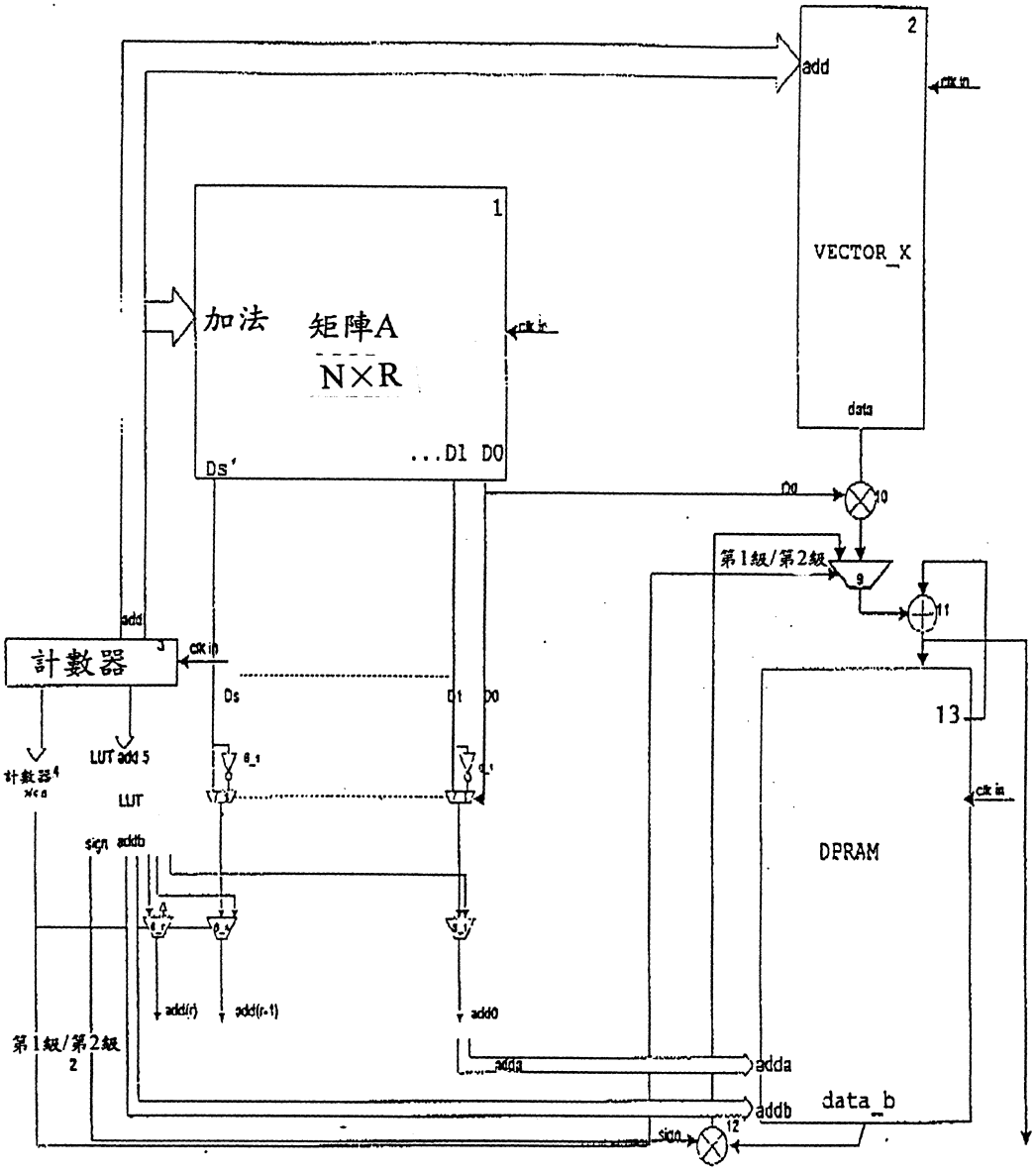
2
add
clk in
VECTOR_X
data
1
加法
矩陣A
N×R
clk in
Ds'
...D1 D0
D0
10
第1級/第2級
9
11
add
計數器
3
clk in
Ds
D1
D0
13
計數器4
LUT add 5
LUT
clk in
DPRAM
sign
addb
add(r)
add(r-1)
add0
第1級/第2級
2
adda
addb
data_b
sign
12

圖 5

修正本 91年3月29日
補充

公告本

A4
C4

中文說明書修正頁(91年3月)

| | |
|---|---|
| 申請日期 | 90.10.5 |
| 案號 | 90124691 |
| 類別 | G06F17/14 |

（以上各欄由本局填註）

# 發明 新型 專利說明書 522316

| | | |
|---|---|---|
| 一、發明新型名稱 | 中文 | 用以有效執行線性轉換之方法及裝置 |
| | 英文 | METHOD AND APPARATUS FOR EFFECTIVELY PERFORMING LINEAR TRANSFORMATIONS |
| 二、發明創作人 | 姓名 | 1.丹尼 葉琳 DANNY YELLIN<br>2.多蘭 雷尼希 DORON RAINISH<br>3.艾弗納 多爾 AVNER DOR |
| | 國籍 | 均以色列 |
| | 住、居所 | 1.以色列拉安納市賀奇爾街71號<br>2.以色列拉梅甘市奇旭街3號<br>3.以色列奇瑞奧諾市哈曼加爾街1818號 |
| 三、申請人 | 姓名（名稱） | 美商英特爾公司 INTEL CORPORATION |
| | 國籍 | 美國 |
| | 住、居所（事務所） | 美國加州聖塔卡拉瓦市米遜大學路2200號 |
| | 代表人姓名 | 湯姆士 C. 雷納德 THOMAS C. REYNOLDS |