COMPUTING SIZE-INDEPENDENT MATRIX PROBLEMS ON SYSTOLIC ARRAY PROCESSORS

Juan J. Navarro
José M. LLaberia
Mateo Valero

Dept. Arquitectura de Computadores
Facultad de Informática (UPC) Pau Gargallo 5.
08028 BARCELONA (SPAIN)

ABSTRACT

A methodology to transform dense to band matrices is presented in this paper. This transformation, is accomplished by triangular blocks partitioning, and allows the implementation of solutions to problems with any given size, by means of contraflow systolic arrays, originally proposed by H.T. Kung. Matrix-vector and matrix-matrix multiplications are the operations considered here.

The proposed transformations allow the optimal utilization of processing elements (PEs) of the systolic array when dense matrix are operated. Every computation is made inside the array by using adequate feedback. The feedback delay time depends only on the systolic array size.

1.- INTRODUCTION

The systolic array processors allow attainment of very high throughput from the high degree of parallelism and pipelining they can support. Most part of the systolic systems so far developed, are tailored to some applications. A particular design is made to meet one (or several related) algorithm(s) and to suit the size of a given data structure size.

The situation in many practical cases is that the number of PEs and the interconnection topology are fixed, but several similar problems with dimensional variations are to be solved in the systolic array processor. In such cases, some transformations of the original data structures are needed.

This work was supported by the Ministery of Education of Spain (CAICYT) under Grant Number 2906-83 C03-03 and by CTNE.

In order to minimize the global computational time it is of great importance:

a) to have data transformations with low generation difficulties,
b) to reach a maximum operations rate in the array, and
c) to get a simple attainment of final results from the partial values computed inside the array system.

Clearly matrix calculations belong to an application field that requires this kind of transformations /1/.

Several authors have focused attention on this problem. K. Hwang and Y.H. Cheng have worked on this direction /2/, proposing matrix partitioning for VLSI arithmetic systems. H.Y. Chuang and G. He presented in /3/ a design methodology of problem size independent systolic array systems, taking into account algorithms without data contraflow. For band matrix operations, good efficiency is achieved with the contraflow systolic arrays proposed by H.T. Kung /4/,/5/, but these systems suffer a throughput decrease when dense matrices are operated. Based upon Kung's design, R.W. Priester and others /6/ present a matrix transformation yielding to a 50% size reduction of the systolic array, with no overhead in the algorithm time under certain conditions.

To evaluate the system efficiency, the utilization factor of each PE in the array, can be used. This measure, η , is expressed as N/AT, where N is the number of operations required by the algorithm, A is the number of PEs in the array, and T is the number of steps needed to execute the algorithm in the system.

In the present work, we propose a method to transform dense matrices of any dimension into band matrices. The transformed matrix may have variable bandwidth, so that an easy and adequate matching to the dimensions of Kung's systolic arrays is achieved. Maximum efficiency is obtained because every array operation cycle is useful, due to the fact that the transformed matrix band is filled (no empty position) with elements from the original matrix. The data transformation is simple enough, and no computation outside the array is needed. Feedback of partial results obtained inside the system, is provided. This fact yields to minimize the algorithm's execution time.

In section 2 two types of data transformation are proposed and developped. We shall name these transformations as **DBT** (Dense to Band matrix transformation by Triangular blocks partitioning). To solve the problem of Matrix-Vector multiplication, an algorithm, **DBT-by-rows**, is proposed in section 2. rAnother algorithm, **DBT-transposed-by-rows**, is used in section 3, to solve the problem of Matrix-Matrix multiplication.

2.- MATRIX-VECTOR MULTIPLICATION.

In what follows, we suppose A to be the original m-by-m matrix and $\bar{A}$ is the transformed band matrix, with bandwith w equal to the array size of the linear matrix-vector multiplication array /5/.

The general method that we are proposing, to map A into $\bar{A}$, is based upon the three following points (see fig.1):

a) To split the original matrix, A(n,m), into $\bar{n}\bar{m}$ submatrices $A_{ij}(w,w)$, where $\bar{n}=\lceil n/w \rceil$ and $\bar{m}=\lceil m/w \rceil$. When n and/or m are not integer multiples of w, A is extended with zero-valued elements in rows and/or columns.

b) Every submatrix $A_{ij}(w,w)$ is, in turn, split into triangular submatrices. Let us call them U_{ij} (upper) and L_{ij} (lower). The main diagonal of A_{ij} may belong to any of them. Let us suppose, without lack of generality, that it belongs to U_{ij}.

c) The resulting band matrix, $\bar{A}$, is formed by submatrices $\bar{U}_k$ and $\bar{L}_k$. We obtain this matrix if submatrices U_{ij} and L_{rs} of a are appended together inside the band.

Several ways to obtain $\bar{A}$ may be devised. Of greater interest will be those leading to raise the efficiency in the global processing of transformation, operation, and attainment of resulting values.

Let us assume that we need to solve the operation y=Ax+b, where A is an n-by-m matrix. The original computation must be accomplished by means of the transformed operation $\bar{y}=\bar{A}\bar{x}+\bar{b}$.

In the fig. 1.a we show a diagram of this operation to be performed, together with the A matrix decomposition into triangular submatrices U_{ij} and L_{ij}, and of the x,b and y vectors. The x, b and y vectors are split into $\bar{m}$, $\bar{n}$ and $\bar{n}$ sub-vectors, respectively, all with w elements.

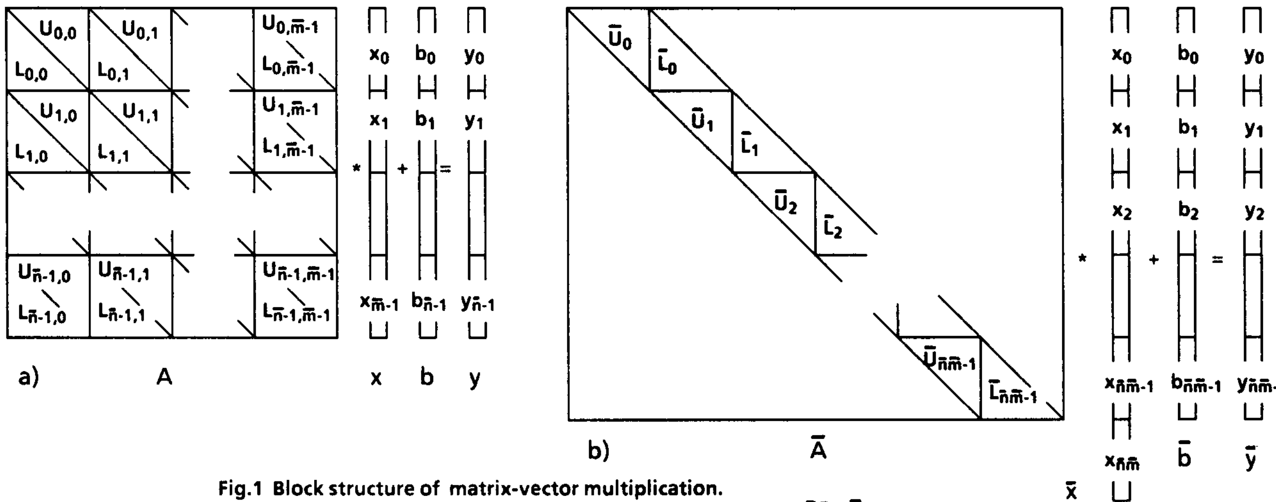


Fig.1 Block structure of matrix-vector multiplication.
a) Original problem Ax + b = y, b) Transformed problem $\bar{A}\bar{x}+\bar{b}=\bar{y}$

The transformation of this problem to one with band matrices can be seen in fig. 1.b. We have assumed, without loss of generality, that the transformed matrix is of the upper-band type ($\bar{A}_{ij} = 0$ for $i>j$). A lower band transformed matrix could be considered in a similar way ($\bar{A}_{ij} = 0$ for $i<j$).

The following conditions are to be satisfied, in order to obtain the required matrix transformation:

1) For $0 \leqslant k < \bar{n}\bar{m}$, if $\bar{U}_k$ is equal to U_{ij}, $\bar{L}_k$ must then be equal to $L_{i,p}$ for any p such that $0 \leqslant p < \bar{m}$.

2) For $0 \leqslant k < \bar{n}\bar{m}-1$, if $\bar{L}_k$ is equal to U_{ij}, $\bar{U}_{k+1}$ must then be equal to $U_{p,j}$ for any p such that $0 \leqslant p < \bar{n}$.

3) Only one single copy of the original submatrices U_{ij} and L_{ij} must exist in the $\bar{A}$ matrix.

The transformed vector, $\bar{x}$, must be composed by $\bar{n}\bar{m}+1$ sub-vectors ($\bar{x}_0$, $\bar{x}_1, \ldots,$ $\ldots$, $\bar{x}_{\bar{n}\bar{m}}$) such that the $\bar{n}\bar{m}$ first of them have w elements and the last one, $\bar{x}_{\bar{n}\bar{m}}$, has w-1 elements. The total number of elements in $\bar{x}$ is $\bar{n}\bar{m}w+w-1$.

The $\bar{b}$ and $\bar{y}$ vectors are formed by $\bar{n}\bar{m}$ sub-vectors, each with w elements. The transformed vectors, $\bar{x}$, $\bar{b}$ and $\bar{y}$, have its structure dependent on the chosen transformation algorithm.

The rules defining the correspondence between the sub-vectors of $\bar{x}$, $\bar{b}$ and $\bar{y}$ and the x_i, b_i, y_i sub-vectors of x, b and y, are as follows:

1) For $0 \leqslant k < \bar{n}\bar{m}$, if $\bar{U}_k = U_{ij}$ then

$$\bar{x}_k = x_j$$

$$\bar{b}_k = \begin{cases} b_i & \text{if } k = \min(I_i) \\ y_i^R & \text{otherwise, where } R = \max(J_i^k) \end{cases}$$

$$\bar{y}_k = \begin{cases} y_i & \text{if } k = \max(I_i) \\ y_i^R & \text{otherwise} \end{cases}$$

I_i and J_i^k are the set of indices

$$I_i = \left\{ q \in \mathbb{N} \mid \bar{U}_q = U_{i,p} \text{ with } 0 \leqslant p < \bar{m} \right\}$$

$$J_i^k = \left\{ q \in \mathbb{N} \mid \bar{y}_p = y_i^q \text{ with } 0 \leqslant p < k \right\}$$

2) If $\bar{L}_{\bar{n}\bar{m}-1} = L_{i,j}$ then $\bar{x}_{\bar{n}\bar{m}} = x'_j$ where x'_j is the sub-vector formed by the (w-1) first elements of the x_j block.

Note that data from the original b_i vector, as well as previously computed partial results, y_i^R, are inputs to the array system. By using this type of feedback, final results are obtained without need of any calculation external to the array processor.

We can see, from the expressions above, that several optimal **DBT** transformations can be devised. We refer to the optimality with regard to the required computational time or EP's utilitation. From those transformations, in this section we choose and present now one that allows simple and regular transformed structures, and with a constant time value of the required feedback. A **DBT-by-rows**, accomplishing the above stated requirements, will be presented in the following paragraphs.

The rules to define such a transformation are as follows:

a) Attainment of $\bar{A}$ from A.

For $0 \leqslant k < \bar{n}\bar{m}$

$$\bar{U}_k = U_{r,s} \text{ with } r = \lfloor k/\bar{m} \rfloor \text{ and } s = k \bmod \bar{m}$$

$$\bar{L}_k = L_{r,s} \quad \text{with } r = \lfloor k/\bar{m} \rfloor \text{ and } s = (k \bmod \bar{m}+1) \bmod \bar{m}$$

b) Attainment of $\bar{x}$, $\bar{b}$, and $\bar{y}$ from x,b, and y.

For $0 \leqslant k < \bar{n}\bar{m}$

$$\bar{x}_k = x_{k \bmod \bar{m}} \quad \text{and} \quad \bar{x}_{\bar{m}\bar{n}} = x'_0$$

$$\bar{b}_k = \begin{cases} b_{k/\bar{m}} & \text{if } k \bmod \bar{m} = 0 \\ y^{k-1}_{\lfloor k/\bar{m} \rfloor} & \text{otherwise} \end{cases}$$

$$\bar{y}_k = \begin{cases} y_{(k+1)/\bar{m}} & \text{if } (k+1) \bmod \bar{m} = 0 \\ y^{k}_{\lfloor k/\bar{m} \rfloor} & \text{otherwise} \end{cases}$$

The PRT transformation proposed by R.W. Priester et al. /6/ is a particular case of the **DBT-by-rows** when $\bar{n}=\bar{m}=1$.

In a **DBT-by-rows**, the number of steps to have the required feedback equals the array size, w, and can be implemented with w registers. This implementation is very modular and easily expandible.

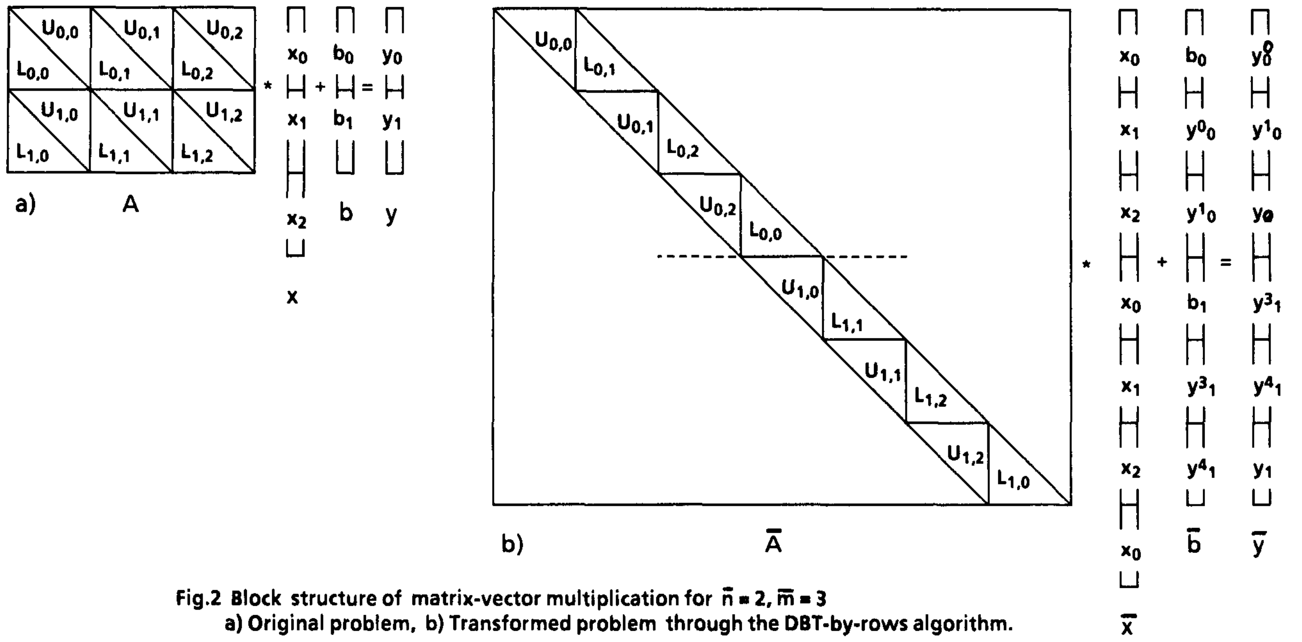


Fig.2 Block structure of matrix-vector multiplication for $\bar{n} = 2, \bar{m} = 3$
a) Original problem, b) Transformed problem through the DBT-by-rows algorithm.

Let us think now of a particular and practical case, with n=6, m=9 and w=3. The block level original data structures can be seen in fig. 2.a; the corresponding transformed data structures are shown in fig. 2.b. In fig. 3 the data coming in and out of the systolic array in every one of the 39 required computational cycles, are shown.

The value of the PE's utilization can be raised 100% by grouping every 2 PEs in 1, or overlapping the execution of several problems, or partitioning the transformed problem into two disjoint sub-problems. The dotted line in fig. 2.b shows the optimal partitioning for the concrete case we are considering.

In a general case, the number of time units required to solve the problem, with no overlapping is: $T = 2w\bar{n}\bar{m}+2w-3$. If overlapping is used, the number of steps is: $T = w\bar{n}\bar{m}+2w-2$.

The processor utilization, η, is given by

$$\frac{n.m}{AT} = \frac{1}{2+\frac{2}{\bar{n}\bar{m}}-\frac{3}{w\bar{n}\bar{m}}} \quad \text{with no overlapping,}$$

and $$\frac{1}{1+\frac{2}{\bar{n}\bar{m}}-\frac{2}{w\bar{n}\bar{m}}} \quad \text{with overlapping.}$$

When the value of the product $\bar{n}\bar{m}$ is large, the PE's utilization aproaches to 1/2 and 1, respectively.

In the next section beside the above presented transformation, another one is to be used. This new transformation, named **DBT-transposed-by-rows**, yields to attainment of a lower-band transformed matrix. The method consists in transposing the matrix resulting from the application of a **DBT-by-rows** transformation to the transposition of the original matrix; that is:

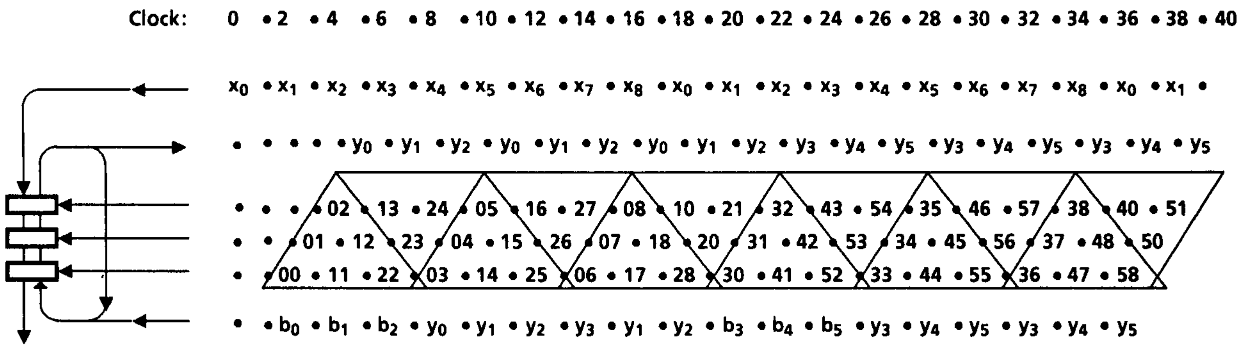
$$\textbf{DBT}_{\textbf{transposed-by-rows}}(A) =$$
$$= \textbf{DBT}_{\textbf{by-rows}}(A^{T})^{T}$$

3.- MATRIX-MATRIX MULTIPLICATION.

In this section we are directed to solve the problem C=A*B, using the w-by-w hexagonal matrix-matrix multiplication array /5/. A, B and C are matrices of (n,p), (p,m) and (n,m) dimensions respectively. $\bar{n}$, $\bar{p}$ and $\bar{m}$ are the coefficients that relate the problem and array dimensions, as follows:

$$\bar{n} = \lceil n/w \rceil, \ \bar{p} = \lceil p/w \rceil \ \text{and} \ \bar{m} = \lceil m/w \rceil$$

To achieve this problem's solutions, the B matrix is divided in column submatrices of width w. The C matrix is attained by succesively multiplying the A matrix by each column submatrix of B matrix. By using this algorithm, the



Clock: 0 • 2 • 4 • 6 • 8 • 10 • 12 • 14 • 16 • 18 • 20 • 22 • 24 • 26 • 28 • 30 • 32 • 34 • 36 • 38 • 40
x0 • x1 • x2 • x3 • x4 • x5 • x6 • x7 • x8 • x0 • x1 • x2 • x3 • x4 • x5 • x6 • x7 • x8 • x0 • x1 •
• • • • y0 • y1 • y2 • y0 • y1 • y2 • y0 • y1 • y2 • y3 • y4 • y5 • y3 • y4 • y5 • y3 • y4 • y5
• • • 02 • 13 • 24 • 05 • 16 • 27 • 08 • 10 • 21 • 32 • 43 • 54 • 35 • 46 • 57 • 38 • 40 • 51
• • 01 • 12 • 23 • 04 • 15 • 26 • 07 • 18 • 20 • 31 • 42 • 53 • 34 • 45 • 56 • 37 • 48 • 50
• 00 • 11 • 22 • 03 • 14 • 25 • 06 • 17 • 28 • 30 • 41 • 52 • 33 • 44 • 55 • 36 • 47 • 58
• • b0 • b1 • b2 • y0 • y1 • y2 • y3 • y1 • y2 • b3 • b4 • b5 • y3 • y4 • y5 • y3 • y4 • y5

Fig.3 Input and output data flow for the problem $\bar{A}\bar{x} + \bar{b} = \bar{y}$ with $\bar{n} = 6$, $m = 9$, $w = 3$.

transformed problem is now $\bar{C}=\bar{A}*\bar{B}$ (see fig. 4.a), and the matrices $\bar{A}$ and $\bar{B}$ are defined as follows:

1) Algorithm to obtain the transformed matrix $\bar{A}$ can be expressed as follows:

   a) To apply a **DBT-by-rows** to matrix A. This step yields to matrix $\bar{A}^b$.

   b) To juxtapose $\bar{m}$ blocks A^b and one triangular block U'. This step yields to matrix $\bar{A}$. The block U' is composed by the first (w-1) rows and (w-1) columns of A^b.

   Consequently, $\bar{A}$ is a square matrix of dimension $\bar{p}\bar{n}\bar{m}+w-1$.

2) The matrix $\bar{B}$ has the same dimension as $\bar{A}$. An algorithm to obtain the transformed matrix $\bar{B}$ can be expressed as follows:

   a) The B matrix is divided into $\bar{m}$ column sub-matrices, with p-by-w elements each. Let us name these sub-matrices by B_0, B_1, ..., $B_{\bar{m}-1}$.

   b) A **DBT-transposed-by-rows** is applied to each B_i. This step yields to a band matrix, B_i^b.

   c) By juxtaposition of $\bar{n}$ blocks B_i^b, the band matrix B_i^d is attained.

   d) By juxtaposition of $\bar{m}$ matrices B_0^d, B_1^d, ..., $B_{\bar{m}-1}^d$ and appending at the end the main sub-matrix L', the $\bar{B}$ matrix is attained. L' is formed by the first (w-1) rows and columns of B_0^b.

Now, we will describe the process of attainment of matrix C, starting from the partial results produced by the systolic array system, when it operates to make the multiplication $\bar{C}=\bar{A}*\bar{B}$.

The partial results matrix, $\bar{C}$, is shown in fig 4, where the C matrix can be also seen. Both matices are decomposed as square w-by-w matrices. Each square matrix is split into three blocks, named U, L and D. The subscript indices notation in $\bar{C}$ reflects its correspondence with matrix C, and the superscript indices express the computational sequence of partial results attainment. To obtain the blocks for matrix C, we must compute:

$$D_{ij}=\sum_{t=0}^{\bar{p}-1} D_{i,j}^{t};\quad U_{i,j}=\sum_{t=0}^{2\bar{p}-1} U_{i,j}^{t};\quad L_{rs}=\sum_{t=0}^{2\bar{p}-1} U_{r,s}^{t}$$

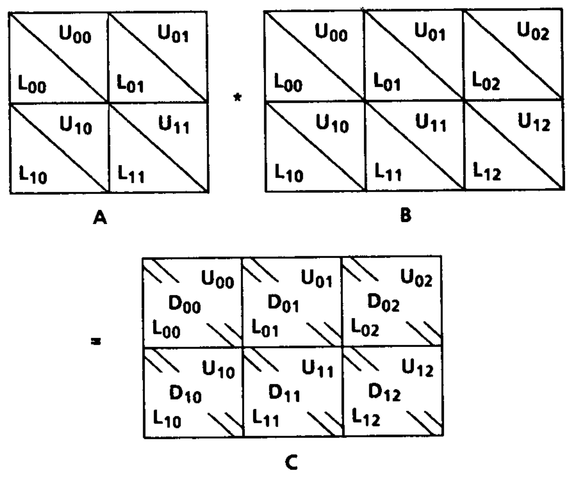


Fig. 4.a Block structure of the original problem AB = C with $\bar{n}=2$, $\bar{p}=2$, $\bar{m}=3$.

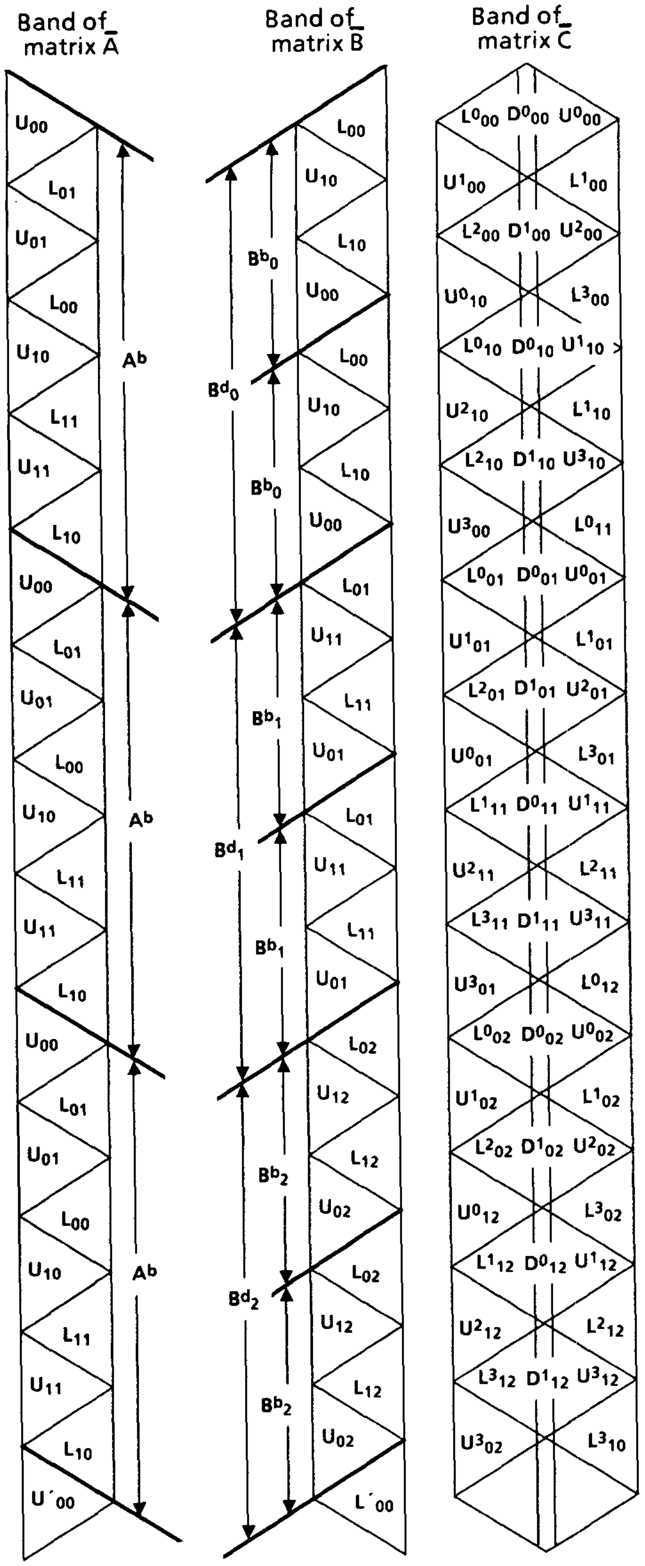

Band of $\bar{\ }$ matrix $\bar{A}$
Band of $\bar{\ }$ matrix $\bar{B}$
Band of $\bar{\ }$ matrix $\bar{C}$
U_{00}
L_{01}
U_{01}
L_{00}
U_{10}
L_{11}
U_{11}
L_{10}
A^b
B^b_0
B^d_0
B^b_1
B^d_1
B^b_2
B^d_2
U'_{00}
L'_{00}
L^0_{00} D^0_{00} U^0_{00}
U^1_{00}
L^1_{00}
L^2_{00} D^1_{00} U^2_{00}
U^0_{10}
L^3_{00}
L^0_{10} D^0_{10} U^1_{10}
U^2_{10}
L^1_{10}
L^2_{10} D^1_{10} U^3_{10}
U^3_{00}
L^0_{11}
L^0_{01} D^0_{01} U^0_{01}
U^1_{01}
L^1_{01}
L^2_{01} D^1_{01} U^2_{01}
U^0_{01}
L^3_{01}
L^1_{11} D^0_{11} U^1_{11}
U^2_{11}
L^2_{11}
L^3_{11} D^1_{11} U^3_{11}
U^3_{01}
L^0_{12}
L^0_{02} D^0_{02} U^0_{02}
U^1_{02}
L^1_{02}
L^2_{02} D^1_{02} U^2_{02}
U^0_{12}
L^3_{02}
L^1_{12} D^0_{12} U^1_{12}
U^2_{12}
L^2_{12}
L^3_{12} D^1_{12} U^3_{12}
U^3_{02}
L^3_{10}
L_{02}
U_{02}
U_{12}
L_{12}

Fig. 4.b Block structure of the transformed problem $\bar{A}\bar{B} = \bar{C}$ with $\bar{n} = 2$, $\bar{p} = 2$, $\bar{m} = 3$.

Fig. 5 shows a feedback topology suited to complete the above computations. In /7/ the systolic arrays with this kind of feedback scheme are named "spiral systolic arrays". The main diagonal is "auto-feedbacked", and the sub-diagonals are feedbacked in pairs, in such a way that the number of processing elements in the loop equals w.

In the computation of U_{ij} and L_{rs}, the feedback delay time equals w, except in some special cases where this delay time is greater than w. These irregularities in the feedback delay time arise because we aim to minimize the computational time required. Nevertheless, a regular delay time can be reached, and therefore a simplification of the control section attained, at the expense of increasing the global computational time. To achieve this goal of regularity, the original problem should be partitioned into $\bar{m}$ subproblems, each subproblem consisting of multiplication of matrix A by every column submatrix of B. The transformed problems, from these subproblems, cannot be linked together without a loss of efficiency; to maintain the calculations, separation of subproblems with zero value blocks is needed /8/.

For the case we are considering, three types of irregularities in the feedback delay time can be recognized. One of these types arise when the blocks U_{0j} are feedbacked. The other two types arise when feedback of blocks $L_{\bar{n}-1,j}$ is used.

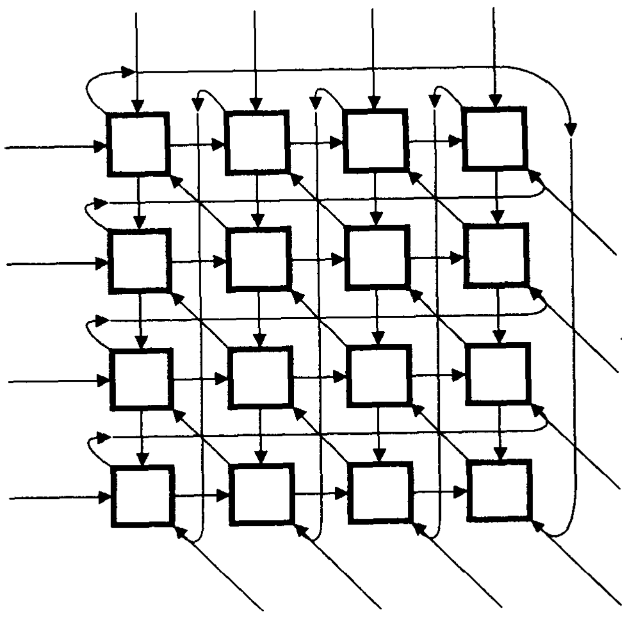


Fig. 5. Interconnection topology of the Hexagonal Systolic Array Processor with spiral feedback.

To calculate blocks $U_{0,j}$, a partial result is first computed:

$$\sum_{t=0}^{2\bar{p}-2} U_{0j}^{t}$$

with a feedback delay time of partial results equals w. The feedback delay of the last partial result is $6(w-1)(\bar{n}-1)\bar{p}+w$.

The calculation proces for $L_{\bar{p}-1,0}$ is the same as for $U_{0,j}$; but now the required feedback delay of the last partial result is $6(\bar{n}\bar{p})(\bar{m}-1)(w-1)+w$.

When calculating $L_{\bar{n}-1,j}$, with $j \neq 0$, the feedback delay of the first partial result is $6(w-1)(\bar{n}-1)\bar{p}+w$. Once the second partial result has been obtained, the following ones are obtained every w cycles.

To use feedback with constant delay time makes necessary a number of 2w and w memory elements, for the main diagonal and sub-diagonals, respectively. With regard to the irregular feedbacks, $w(w-1)3/2$ memory elements are needed.

Taking into account these features, as well as the feedback topology, a modular and easily expandible design is possible for the systolic array system.

The number of steps required to solve the problem is: $T = 3w\bar{p}\bar{n}\bar{m} + 4w - 5$. The processor utilization is given by

$$\frac{n.m.p}{AT} = \frac{1}{3 + \frac{4}{\bar{p}\bar{n}\bar{m}} - \frac{5}{w\bar{p}\bar{n}\bar{m}}}$$

A formal description of the data flow that must feed the array, can be found in the Appendix of this paper. The input of data must be according to the diagonals direction, to attain the computation of C=A*B+E without need of any operation outside the array. The description of the data flow is made and presented at a D, L and U blocks level.

4.- CONCLUSIONS

A matrix structure transformation method has been presented in this paper. The aim is to use, with a maximum efficiency, the fixed size systolic arrays proposed by H.T. Kung, for arbitrarily sized matrices. The method consists in the transformation of a dense matrix, A, into a band matrix, $\bar{A}$; this band matrix has a bandwidth equal to the array size. To achieve this goal, A is split into triangular sub-matrices. Later, a general juxtaposition algorithm (**DBT**) for triangular sub-matrices, is used, and the transformed matrix is obtained.

Matrix transformation algorithms that have been presented in this paper are of two kinds: **by-rows** and **transpose-by-rows**. These algorithms allow utilization of systolic arrays without any efficency loss when solving problems of the classes matrix-vector and matrix-matrix multiplication. The number of memory elements required depends only on the array size. The topology of feedback flows is regular and fixed for a given system. Moreover, all the computations are made inside the systolic system, and modular, expandible structures are adequate to this purpose.

From the proposed transformations, some other related types of transformations are easily deduced, and simplification of the feedback scheme are attained, at the cost of a lower optimization of the PE's utilization

In the case of computing with matrices of a known degree of sparsity, transformation algorithms can be devised and developed, to exclude the need of zero-valued elements sub-matrices. A reduction of computational time would be the consequence of using such algorithms.

The methodology that has been presented in this paper has been also applied to solve the problems: Triangular systems of linear and matrix equations, Gauss-Seidel iterative method, L-U decomposition and inverses of triangular and dense matrices /8/. Other types of matrix algorithms are presently in study, and the hardware design of a prototype system is under execution.

APPENDIX

The flow of data that must feed the array system to accomplish the computation C=A*B+E, without any external operation, is expressed now.

A band matrix with a value of width equal to 2w-1, is formed from the data introduced through the array diagonals; let us name this matrix as I (input). These data are partialy processed inside the array system, and another matrix is formed at the array output; let us name this matrix as O (output). We divide these two matrices, I and O, into square sub-matrices with w-by-w elements, and each one of these sub-matrices is again

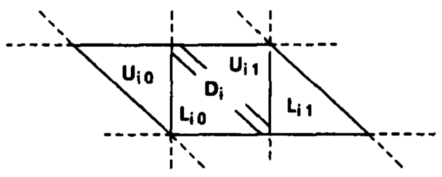


Fig. 6 Notation of row block i for matrices I (Input) and O (Output).

divided into lower triangular blocks (L), diagonal blocks (D), and upper triangular blocks (U).

To denote the sub-matrices in the band of every row of blocks, subscript indices are used as shown in Fig. 6. Each sub-matrix name receives one superscript index, denoting the matrix to which it belongs.

A input matrix I, is formed from the data matrix E and from the feedbacked array output, O.

Matrix I can be composed as follows:

For $0 \leqslant k \leqslant \bar{n}\bar{m}$

$$U^I_{k,0} = \begin{cases} U^O_{k-\bar{p}(\bar{n}-1)-1,1} & \text{if } k \bmod \bar{p}\bar{m} = 0 \\ U^E_{k/\bar{p} \bmod \bar{n},\lfloor k/\bar{p}\bar{n}\rfloor} & \text{if } k \bmod \bar{p} = 0 \text{ and } k \bmod \bar{p}\bar{n} \neq 0 \\ U^O_{k-1,1} & \text{otherwise} \end{cases}$$

$$L^I_{k,0} = \begin{cases} L^O_{k-\bar{p}(\bar{n}-1)-1,1} & \text{if } (k+\bar{p}) \bmod \bar{p}\bar{n} = 0 \text{ and } k \neq \bar{p}(\bar{n}-1) \\ L^E_{k/\bar{p} \bmod \bar{n},\, k/\bar{p}\bar{n}} & \text{if } k \bmod \bar{p} = 0 \text{ and } (k+\bar{p}) \bmod \bar{p}\bar{n} \neq 0 \\ L^O_{k-1,1} & \text{otherwise} \end{cases}$$

$$D^I_k = \begin{cases} D^E_{k/\bar{p} \bmod \bar{n},\lfloor k/\bar{p}\bar{n}\rfloor} & \text{if } k \bmod \bar{p} = 0 \\ D^O_{k-1} & \text{otherwise} \end{cases}$$

$$U^I_{k,1} = \begin{cases} U^E_{0,\, k/\bar{p}\bar{n}} & \text{if } k \bmod \bar{p} = 0 \\ U^O_{k,0} & \text{otherwise} \end{cases}$$

$$L^I_{k,1} = \begin{cases} L^O_{\bar{p}\bar{n}-1,0} & \text{if } k = \bar{p}\bar{n}\bar{m}-1 \\ L^E_{\bar{n}-1,\,(k+1)/\bar{p}\bar{n}} & \text{if } (k+1) \bmod \bar{p}\bar{n} = 0 \text{ and } k \neq \bar{p}\bar{n}\bar{m} \\ L^O_{k,0} & \text{otherwise} \end{cases}$$

The submatrices from the result matrix C are attained, at the array output, as follows:

For $0 \leqslant i, < \bar{n} \quad 0 \leqslant j < \bar{m}$

$$L^C_{i,j} = \begin{cases} L^O_{\bar{n}\bar{p}\bar{m}-1,1} & \text{if } (i,j) = (\bar{n}-1,0) \\ L^O_{(j+1)\bar{p}\bar{n}-1,0} & \text{if } i = \bar{n}-1,\ j > 0 \\ L^O_{(i+j\bar{n}+1)\bar{p}-1,1} & \text{otherwise} \end{cases}$$

$$D^C_{i,j} = D^O_{(i+j\bar{n}+1)\bar{p}-1}$$

$$U^C_{i,j} = \begin{cases} U^O_{(j+1)\bar{p}\bar{n},0} & \text{if } i = 0 \\ U^O_{(i+j\bar{n}+1)\bar{p}-1,1} & \text{otherwise} \end{cases}$$

REFERENCES.

/1/ C.L. Seitz, "Concurrent VLSI Architectures", IEEE Trans. on Computers, Vol. C-33. N 12, Dec. 1984, pp. 1247-1265.

/2/ K. Hwang and Y.H. Cheng, "Partitioned Matrix Algorithms for VLSI Arithmetic Systems,". IEEE Trans. on Comp., Vol. C-31. No. 12, Dec. 1982 p.p.1215-1224.

/3/ H.Y. Chuang and G. He, "A Versatile Systolic Array for Matrix Computations", 12th. Anual Int. Symp. on Comp. Arch.1985. Conf. Proceedings, pp. 315, 322.

/4/ H.T. Kung and C.E. Leiserson, "Systolic Arrays (for VLSI)", in I.S. Duff and G.W.Stewart (editors), Sparse Matrix Proceedings 1978,Soc. for Ind. and App. Math., p.p. 256-282, 1979.

/5/ H.T. Kung and C.E. Leiserson. Sec. 3, Chapt. 8 of C. Mead and L. Conway, Introduction to VLSI Systems, Addison-Wesley, 1980.

/6/ R.W. Priester, H.J. Whitehouse, K.Bromley, J.B. Clary, "Signal Processing with Systolics Arrays," Proc. of the 1981 Int. Conf. on Parallel Processing 1981, pp. 207-215.

/7/ S.Y. Kung, "VLSI Array Processors," IEEE ASSP Magazine, July 1985,pp.4-22.

/8/ J.J. Navarro, J.M. Llaberia, M. Valero "Solving Matrix Problems with no Size Restriction on a Systolic Array Processor". Report RR86/01. Facultad de Informática de Barcelona (U.P.C.). Jan. 1.986.