Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE

PISA

VLSI IMPLEMENTATION OF FAST SOLVERS FOR BAND LINEAR SYSTEMS WITH CONSTANT COEFFICIENT MATRIX

B. Codenotti, G. Lotti, F. Romani

Nota interna B84-11

Agosto 1984

VLSI IMPLEMENTATION OF FAST SOLVERS FOR BAND LINEAR SYSTEMS WITH CONSTANT COEFFICIENT MATRIX

B. CODENOTTI[1], G.LOTTI[2] and F. ROMANI[1]

[1]Istituto di Elaborazione dell'Informazione - CNR, via S.Maria 46, PISA, ITALY.

[2]Dipartimento di Scienze dell'Informazione - University of Pisa, Corso Italia 40, PISA, ITALY.

KEY WORDS. VLSI Models, Area-Time Complexity, Elliptic-Equations, Linear Transforms

1. INTRODUCTION

In this paper, some special linear transformations of a vector $\underline{x}$ by a matrix A with constant entries are studied. The most significant example in this class is probably the DFT, for which several optimal VLSI designs have been obtained [BIL,THO]. The transforms here considered involve the inverses of fixed (2k+1)-diagonal matrices.

Such transformations solve the associated band linear systems which often arise from the discretization of boundary value problems by difference methods. As an example, consider the one dimensional problem:

$$-f'' = g \quad \text{on} \quad (a,b),$$

(1.1) $f(a)=c,$

$f(b)=d;$

which leads to the system

(1.2) $A\,\underline{x} = \underline{b},$

where

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & \\ 0 & & \cdots & & 0 \\ \cdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix},$$

and $A^{-1} = B = (b_{ij}),\qquad b_{ij} = \begin{cases} \dfrac{i(n-j+1)}{n+1}, & i \geq j \\[2ex] \dfrac{j(n-i+1)}{n+1}, & j > i. \end{cases}$

The special structure of the inverses of band matrices allows detecting VLSI layouts which can be splitted into two parts with reciprocal low flow of information. Namely a recursive VLSI design is derived requiring a time

$T = 2\log n + 2\log k + 2$ and an area

$A = O(n\,k\,\log^2 n\,\log k)$ to solve the problem.

2. THEORETICAL RESULTS

In this section, we prove some results about the rank of certain minors of the inverse of a band matrix.

PROPOSITION 2.1

Let A be an nxn matrix partitioned as

$$A = \begin{bmatrix} U & V \\ W & Z \end{bmatrix},$$ where U and Z are square matrices of size pxp and (n-p)x(n-p) respectively. Moreover, assume A, U, Z to be nonsingular. Then the following relations hold

$$A^{-1} = \begin{bmatrix} C^{-1} & X \\ Y & D^{-1} \end{bmatrix}, \quad \text{where}$$

$$C = U - V Z^{-1} W, \qquad D = Z - W U^{-1} V,$$

$$C^{-1} = U^{-1} + U^{-1} V D^{-1} W U^{-1}, \qquad D^{-1} = Z^{-1} + Z^{-1} W C^{-1} V Z^{-1},$$

$$X = -U^{-1} V D^{-1} = -C^{-1} V Z^{-1}, \qquad Y = -Z^{-1} W C^{-1} = -D^{-1} W U^{-1}.$$

The proof follows by a straightforward application of LDR block factorization and Woodbury formula [HCU pp. 123-127].

∎

In the following, we assume A to be a (2k+1)-diagonal matrix (i.e. $A = (a_{ij})$, $a_{ij} = 0$, $|i-j|>k$)

PROPOSITION 2.2

Let A be factored as in Proposition 2.1, with the conditions $p \geq k$, $n-p \geq k$. Then the following properties hold

C, D are $(2k+1)$-diagonal matrices,

X, Y have rank k.

∎

The solution $A^{-1}\underline{b}$ of the linear system $A\,\underline{x} = \underline{b}$ can be written as follows.

Let $\underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix}$, $\underline{b} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \end{bmatrix}$, where $\underline{x}_1$, $\underline{b}_1$ are p-vectors and $\underline{x}_2$, $\underline{b}_2$ are $(n-p)$-vectors. Then

$$(2.1)\qquad A^{-1}\underline{b} = \begin{bmatrix} C^{-1}\underline{b}_1 + X\,\underline{b}_2 \\ D^{-1}\underline{b}_2 + Y\,\underline{b}_1 \end{bmatrix} = \begin{bmatrix} C^{-1}\underline{b}_1 + E\,F^T\underline{b}_2 \\ D^{-1}\underline{b}_2 + G\,H^T\underline{b}_1 \end{bmatrix},$$

where X, Y are respectively $p\times(n-p)$ and $(n-p)\times p$ matrices, E, H are $p\times k$ matrices, F, G are $(n-p)\times k$ matrices and

$$E\,F^T = X\,, \quad G\,H^T = Y\,.$$

3. RECURSIVE VLSI DESIGN

Assume that, in our VLSI implementation, multiplication and addition require one time unit and let $n = k\,2^p$.

Formula (2.1) leads to the recursive design of fig.1, where modules of kind I implement the transform of size $n/2$, modules of kind II and III perform the matrix-vector product of size $k \times n/2$ and $n/2 \times k$ respectively. These modules and the

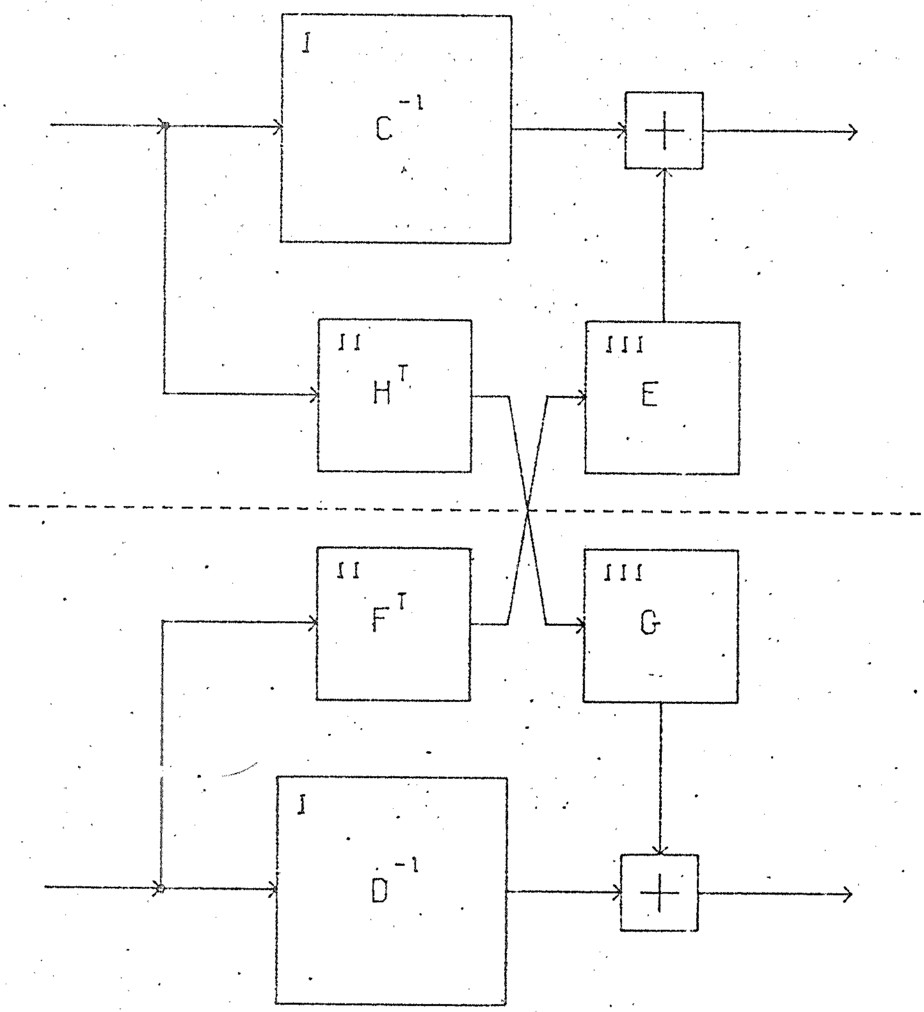

Fig.1 Recursive module structure.

transform of size k use the well known structure of mesh of trees. A pxq matrix-vector product can be performed by a mesh of trees of height $H=O(p \log q)$ and base $L=O(q \log p)$ in $\log p + \log q + 1$ time units [LEI].

Note that the flow of information across the dashed line in fig. 1 is 2k.

PROPOSITION 3.1

Let $t(n)$ be the time required by the algorithm to solve a problem of size n. Then

$$t(n) = 2 \log n + 2 \log k + 2 = O(\log n).$$

Proof.

Let n=k; we use the kxk mesh of trees module followed by $2 \log k + 1$ delay units to get $t(k)=4 \log k + 2$. Assume

$$t(n) = 2 \log n + 2 \log k + 2$$; then it is easy to see that

$$t(2n) = \text{MAX} [\ t(n),\ 2 \log n + 2 \log k + 2\ ] + 2 =$$
$$= 2\log 2n + 2\log k + 2,$$

and the thesis is proved by induction.

∎

A more accurate design is presented in fig.2. The size of this layout can be estimated by simple considerations. In fact, it is easy to see that

$H(k) = O(k \log k)$, from which

$$H(2n) = \text{MAX} [2\ H(n) + n,\ O(n \log k) + n + 2k,\ 2n].$$

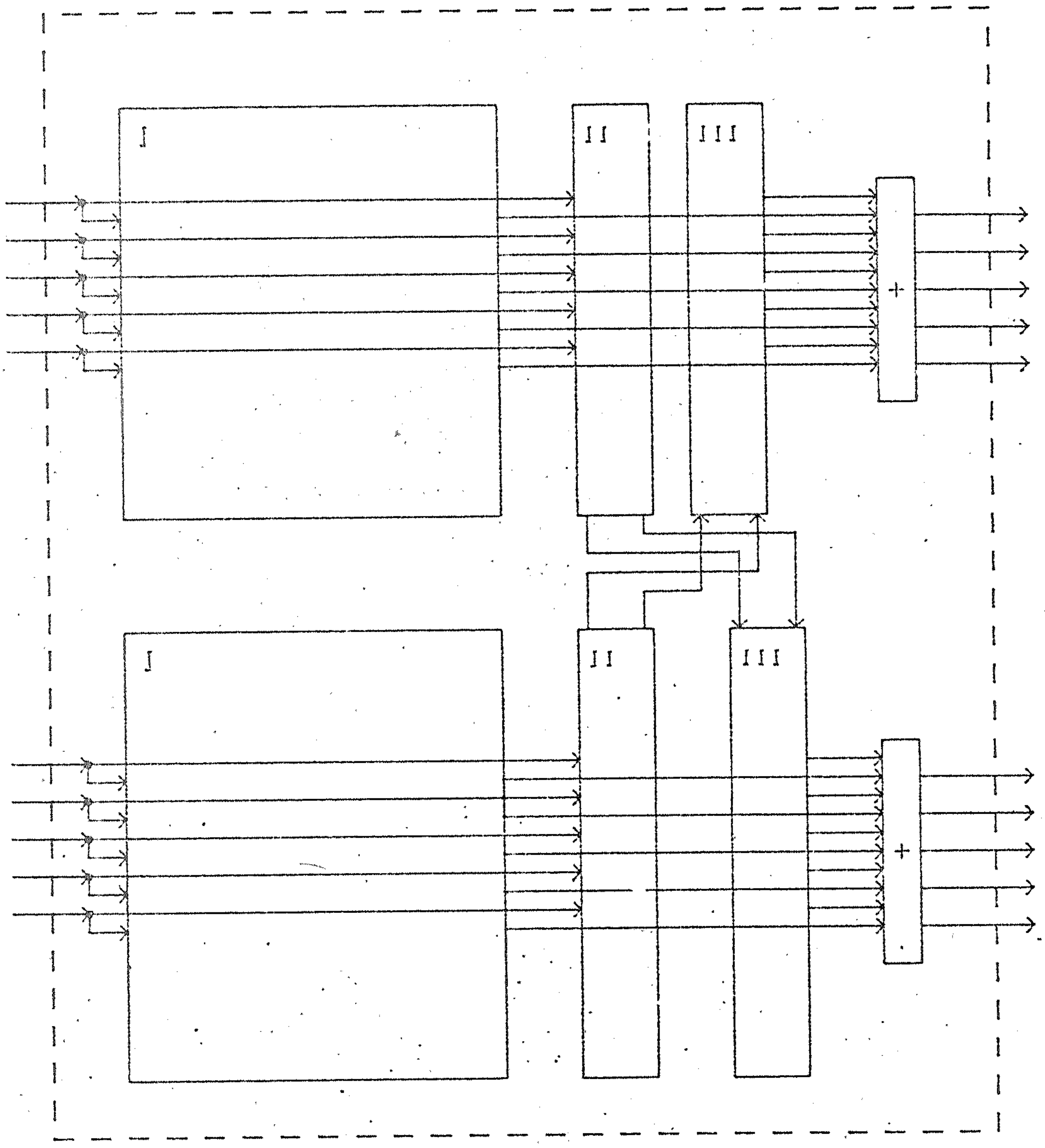

Fig. 2. Recursive VLSI design.

Then, by induction, the bound $H(n) = O(n \log k)$ can be obtained. Analogously, from $L(k) = k \log k$, we get

$$L(2n) = L(n) + 2 k \log n + k + 2,$$

that yields the bound

$$L(n) = \begin{cases} O(k \log^2 n) & \text{if } k = O(\log^a n), \\ O(k \log n) & \text{if } k = O(n^a). \end{cases}$$

Hence, we get

$$AT^2 = \begin{cases} O(n k \log^4 n \log k) & \text{if } k = O(\log^a n), \\ O(n k \log^3 n \log k) & \text{if } k = O(n^a). \end{cases}$$

The case $k=1$ (tridiagonal systems) deserves a special treatment. It is easy to see that

$$T = \log n + 2,$$

$$H = O(n),$$

$$L = O(\log^2 n),$$ from which we get

$$AT^2 = O(n \log^4 n).$$

A possible application of this design is the solution of the linear system (1.2).

4. APPLICATIONS

Any linear system with constant band coefficient matrix can be solved using the above tecniques. In particular, the discretization of an elliptic partial differential equation on a rectangular domain often leads to linear systems with $O(n^2)$ unknowns and a square coefficient matrix of size $O(n^2)$ and an

$O(n)$ band.

An interesting special case is the Poisson equation in a rectangular region [SWA]. The application of Fourier techniques to this problem can be implemented in VLSI with the same complexity of an $O(n) \times O(n)$ bidimensional DFT, yielding the bound $AT^2 = n^4 \log^4 n$ [BIL].

Straightforward application of our algorithm yields the bound $AT^2 = O(n^3 \log^4 n)$, for any differential problem leading to a system with $O(n^2)$ unknowns and $O(n)$ band matrix.

For the Poisson problem it is possible to derive an "ad hoc" VLSI design of simple structure. The discrete approximation of Poisson equation on a uniform grid of size $m \times n$ with various types of boundary conditions, leads to a linear system with mn unknowns. The associated matrix is $m \times m$ block sparse and strongly structured. The unknowns and the right hand terms of the system can be conveniently arranged to form two $m \times n$ matrices, say X and B respectively. A common direct method to solve this problem [SWA] consists of applying to the rows of B a Fourier transform (implemented with the FFT algorithm) yielding an intermediate matrix C.

Then n tridiagonal Toeplitz systems are solved: their right hand terms are the columns of C and the $m \times m$ coefficient matrices T_p have the form:

$$T_p = \begin{bmatrix} q_p & -1 & 0 & \dots & 0 \\ -1 & q_p & -1 & \dots & \\ 0 & & \dots & & 0 \\ & \dots & -1 & q_p & -1 \\ 0 & \dots & 0 & -1 & q_p \end{bmatrix},$$ q_p depending on the type of boundary conditions.

The inverse of T_p has the form:

$$T_p^{-1} = (t'_{ij}), \quad t'_{ij} = \begin{cases} S_{i-1}(q_p)\, S_{n-j}(q_p) / S_n(q_p), & i \geq j, \\ S_{j-1}(q_p)\, S_{n-i}(q_p) / S_n(q_p), & i < j, \end{cases}$$

where $S_i(x)$ is the i-th Chebyshev polinomial of first kind [BAN].

Let D be the matrix of the solutions of the tridiagonal systems; the solution X of the problem is finally obtained by applying to the rows of D the inverse of the previously used Fourier transform.

This algorithm can be implemented in VLSI, according to the layout presented in fig.3. The input data is the matrix B arranged by rows; each row is transformed by an FFT module (S) giving a row of the intermediate matrix C. The columns of C are then processed by n modules (I) solving the associated tridiagonal systems. The inverse trasforms are performed by n modules (T). The constants for the FFT and for the solution of Toeplitz tridiagonal systems are present at start time in the layout.

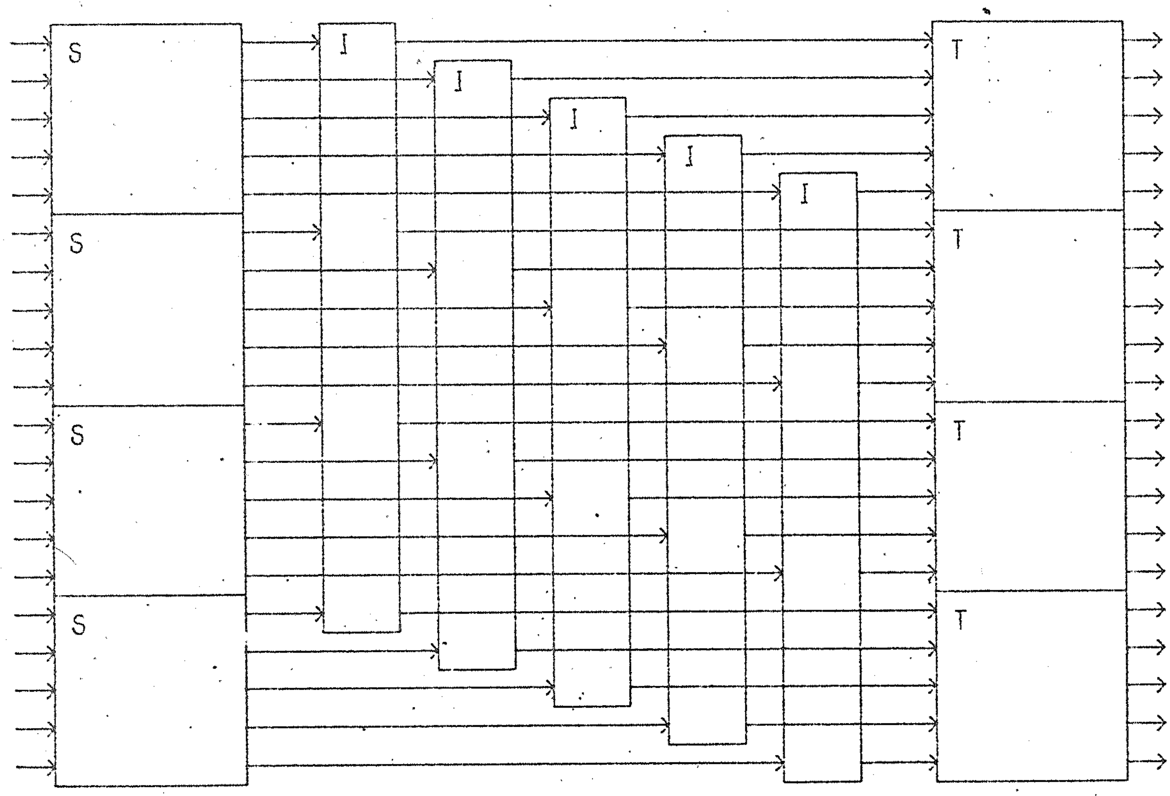

Fig. 3. VLSI Poisson solver.

REFERENCES

R. E. BANK and D.J. ROSE. Marching Algorithms for Elliptic Boundary Value Problems. 1: The Constant Coefficient Case. SIAM J. Numer. Anal. 14(1977) 792-828.

G. BILARDI and M. SARRAFZADEH, Optimal Discrete Fourier Transform in VLSI. International Workshop on Parrallel Computing and VLSI. Amalfi (1984).

A.BOJANCZYK, R.P.BRENT and H.T.KUNG, Numerically Stable Solution of Dense Systems of Linear Equations Using Mesh-Connected Processors, SIAM J. Sc. Stat. Comp. 5 (1984) 95-104.

W.W.BARRET a P.J.FEINSILVER. Inverses of band Matrices. Linear Algebra Appl. 41(1981) 111-130.

F.T.LEIGHTON, New Lower Bound Tecniques for VLSI. Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science (1981), 1-12.

F.P.PREPARATA and J.E.VUILLEMIN, Area-Time Optimal VLSI Networks for Multiplying Matrices. Inf. Proc. Lett. 11 (1980), 77-81.

P.N. SWARTZTRAUBER, The Methods of Cyclic Reduction, Fourier Analysis and the FACR Algorithm for the Discrete Solution of Poisson's Equation in a Rectangle, SIAM Review, 19, (1977) 490-501.

C.D. THOMPSON, Area-Time Complexity for VLSI. Proc. 11th Annual ACM Symp. on the Theory of Computing (SIGACT) (1979) 81-88.