CONICAL SYSTOLIC ARRAY FOR MATRIX INVERSION

H.S.Shahhoseini, A.Khayatzadeh, M.Naderi

Electrical Engineering Department
Iran University of Science and Technology
Tehran Narmak 16844, Iran.
Email: h_s_shahhoseini@hotmail.com

Abstract:

Matrices have been used in many analytical and simulation models and numerical solutions. Matrix operations have essential role in many scientific and engineering applications. One of the most time-consuming operations among matrix operations is matrix inversion. Many hardware designs and software algorithms have been proposed to reduce the time of computation. They will be more important for the large size matrix.
In this paper a systolic array structure is proposed for matrix inversion, which is called Conical Systolic Array, CSA. Gaussian elimination algorithm is used in the CSA. Comparing previous designs, the number of processing elements, PEs, for CSA is 25% less than the previous design yet the total computation time is lower than the previous array.

Keywords: Systolic array, Matrix inversion, Parallel processing, Performance evaluation.

1. Introduction:

Matrices have been used in many areas of engineering and scientific applications for making analytical models. Many algorithms and numerical solutions have been constructed on the basis of matrices in different applications such as signal processing [1], power distribution [2,3], circuit design [4,5], parallel processing [6], image processing [7] and so on. Among matrix operations, matrix inversion is a time-consuming operation, and has been needed in almost all of the above applications. Since the time of operation for matrix inversion is nonlinearly increased by the size of matrix, different software and hardware methods have been proposed to reduce the time of the operation [8-10]. In both software and hardware methods parallelism is key point for reaching the minimum possible time.
Systolic array is the best parallel architecture, which uses fine grain parallelism and can map any recursive algorithms such as matrix inversion algorithm onto VLSI devices [8,9,11]. In this paper a systolic array is designed for matrix inversion. The end of the array has been connected to its beginning, so it is called Conical Systolic Array or CSA. The main contribution of CSA is the reduction of PEs compared to the previous arrays [8,11]. Gaussian elimination without pivoting [9] is applied in CSA. The rest of paper is organized as follows: In section II previous methods for matrix inversion are reviewed. In section III the architecture of CSA and functional tasks of PEs are presented. In section IV the performance of array, on the basis of computation time and chip area, is evaluated. The paper is concluded in section V.

2. Matrix Inversion Methods:

Two main algorithms that have been mostly used for matrix inversion are:
a) L-U decomposition.
b) Gaussian elimination.

In L-U method the $N \ x \ N$ matrix, A, is decomposed into L and U, which are lower and upper triangular matrices respectively, so as $A = L.U$. In second step, $V = L^{-1}$ and $W = U^{-1}$ are computed and in third step A^{-1} is computed as A^{-1} = W.V [8].
In Gaussian elimination {[A | I]} is constructed, in which I is Unity matrix. Then on both sections (A and I), it must be tried that A will be converted to Unity matrix, by Gaussian operation. Then matrix I will be changed to A^{-1}. This algorithm can be done by solving n linear equations, where n is the dimension of A.
Both methods can be done by software or implemented by hardware. Usually hardware solutions are faster than software methods. Among hardware solutions, due to recursive computation, the systolic array can be the best choice. Although systolic array is fast but for large sized problem the array dimension will be very large, and many problems such as clock propagation and difficulty of synchronization will appear. Examples for the applications in which the size of matrix is large, are heart field simulation algorithm where n>2000, [12] or samples of a radar signal for large windows [1]. Currently the software methods are usually used in such applications. To overcome the hardware drawbacks, the reduction of the total number of PEs in the systolic arrays can be key point for reaching the more efficient solution in a large size problem.

3. CSA Architecture

In this section a new systolic array that has conical structure is introduced. The Gaussian elimination without pivoting is used to invert the matrix by the array. The proposed array is called Conical Systolic Array or CSA.
Fig. 1 shows an array for matrix triangularization with n=3. In this array the elements of the reference matrix are injected from top and the intermediate results stay in the processing elements. This array is a bidirectional array and contains three types of PEs. Each PE works in two modes. When the array computing the triangular matrix all PEs work in mode one and then for computing matrix inversion the PEs work in mode two. These two modes are illustrated in Fig. 2.
Fig. 3 shows an array that can convert the element of unity matrix, (which augment to the A). In first mode, the results of triangularization equations are applied to the element of unit matrix that have been generated by square PEs, and in second mode the final elements of unit matrix will be pumped to the triangular array, column by column.

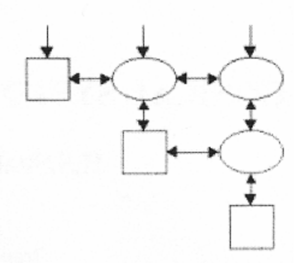

Fig. 1. Array structure that used in first step for matrix triangularization and in second step generates the element of inverse matrix.

In second mode the triangular array receives the columns of converted Unity Matrix and computes the column of inverse matrix. The elements of inverse matrix are sent out from the ports that the elements of reference matrix have been pumped into in the array. Fig. 3 shows the array in which one PE works in mode 1 when the others work in mode 2. In Fig 2 first all PEs work in mode 1 and then in mode 2.

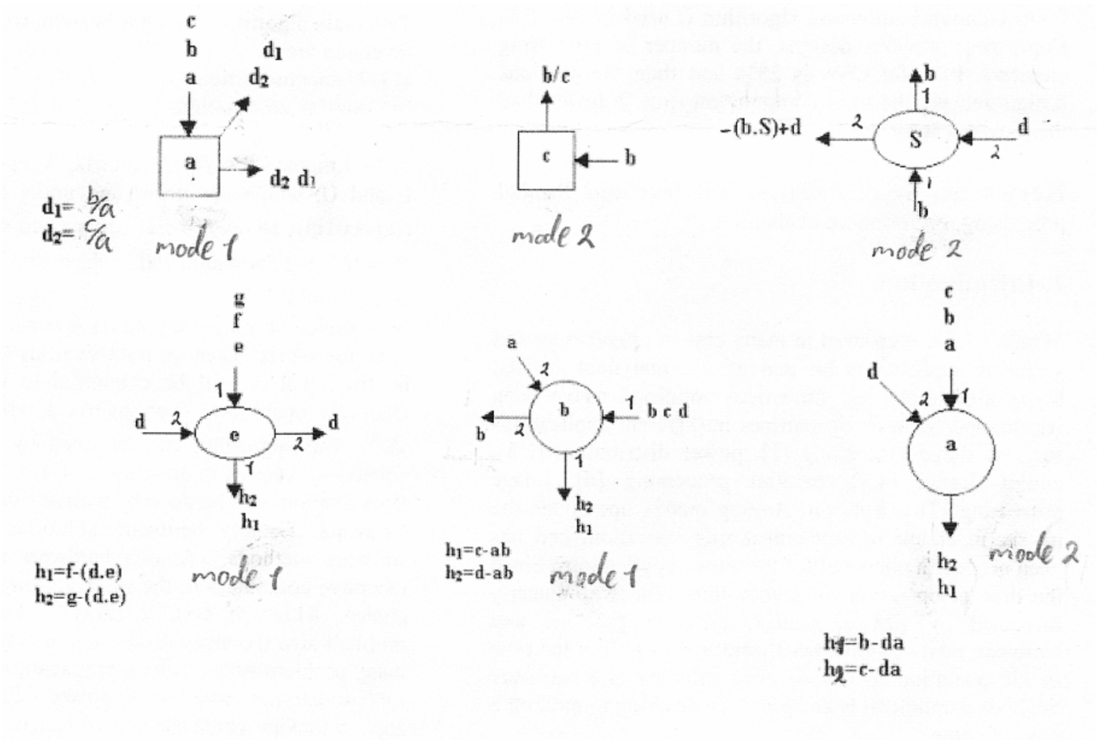

Fig. 2. Two modes of operation for each processing element

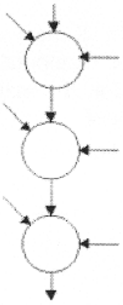

Fig. 3. Array processor for changing the elements of unit matrix

The final array, CSA, and the data sequence for the computation of the matrix inversion are depicted in Fig. 4. The array has conical structure hence called CSA. For simplicity Fig. 4 shows the CSA for *n=3*, and certainly the structure can be easily expanded to any value of n.

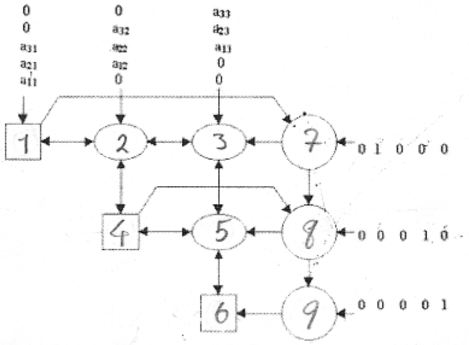

Fig. 4. Total structure of CSA for computing inverse matrix

At (3n-2) time-units the upper/lower triangular matrix is generated but I' is not complete. After triangularization, inverse matrix generation and completing I' are done simultaneously. Inversion is completed after completing I'. The last step of inversion starts from the latest processing element of array in Fig. 2. To complete this step, (3n-2) time-units are needed.

4. Performance Evaluation:

In this section the performance of CSA is evaluated and it is compared with two other previously designed arrays [8,11]. Let's consider CSA needs *D* time-units to compute the inverse matrix.

$$D = D_1 + D_2$$

where D_1 and D_2 are the triangularization delay, and the time needed for solving n linear equation, respectively. So:

$$D = (3n - 2) + (3n - 2) = 6n - 4$$

Comparing previous array, CSA has higher computation time. This array contains of $\frac{n(n+1)}{2}+n$ simple processing elements that arranged as cone form. In Fig. 5 the number of PEs in CSA is compared with the array l that was proposed in [l I], and array 2 that was presented in [8]. It can be seen that CSA has the lowest number of PEs for different size of matrix.

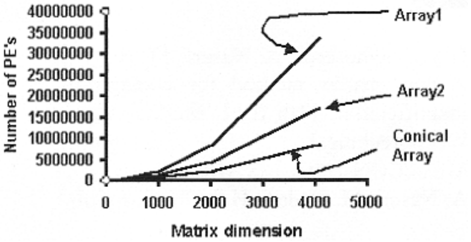

Fig. 5. The number of the processing element in CSA, Array l and Array 2

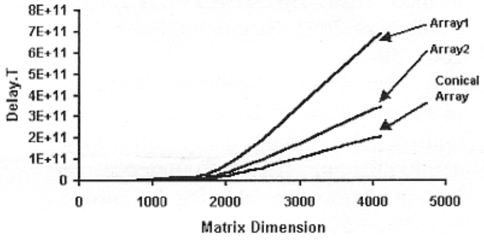

Fig. 6. The total computation time,T_{total}, versus matrix dimension, n, **for CSA, Array l, Array 2.**

To find the total computation time, T_{total} the number of clock cycle, D , must be multiplied by clock period, *T* . T is proportional to the size of the array. When the size of array is increased the clock propagation delay within the chip will be increased and for synchronization the period of clock pulses must be increased. So in a simple model we can assume that clock period can be found by following equation:

$$T = a.N$$

where *N* is the number of elements in the array and *a* is a constant regarding to design technology of array. Fig. 6 illustrates the total computation time, T_{total} ,versus matrix dimension, n , for CSA, Array l and Array 2.

Fig. 5 and Fig. 6 show that CSA has lower number of PEs, i.e., 75% less than Array l and 50% less than Array 2 and also its computation time is lower than Array l and Array 2.

5. Conclusion:

In this paper a systolic design is proposed for matrix inversion. In this array the total structure is triangular but since the end the array is connected to its beginning, it is called Conical Systolic Array or CSA. The communication between PEs is local and has been done in one direction. Matrix inversion by CSA does not need any control signals and its data stream is completely regular. CSA uses Gaussian elimination without pivoting and completes the inversion in 6n-4 steps. Its total number of the processing elements is {n(n+1)/2+n} which is lower than previous design.

References

[I] H.S. Shahhoseini, A. Naseri, M. Naderi, A new matrix method for pulse train identification, 1 1 th IEEE Mediterranean Electrotechnical Conference, MELECON 2002, 183 -187.

[2] A. Naseri, M. Naderi, H.S. Shahhoseini,

Conference, 2002,Dec 2002 - .

[3] J. Kim, E. Matoglu, J. Choi, M. Swaminathan, Modeling of multi-layered power distribution planes including via effects using transmission matrix method, IEEE ASP-DAC/VLSI Design Conference 2002, Bangalore, India, Dec. 2002, 59-65.

[4] R.W. Freund, P. Feldmann, Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Pade approximation, IEEE Conference on Design, Automation and Test in Europe, Feb. 1998, 530-537.

[5] R.Doallo, B. B. Fraguela, E. L. Zapata, Direct mapped cache performance modeling for sparse matrix operations, IEEE 7th Euromicro Workshop on Parallel and Distributed Processing Conference, PDP99, Portugal, 1999, 331-338.

[6] P. Sadayappan, V. Visvanathan, Circuit simulation on shared-memory multiprocessors, IEEE Transactions on Computers, 37(12), Dec.1988, 1634-1642.

[7] I.K.Park, I.D.Yun, S.U.Lee, Models and algorithms for efficient color image indexing, Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries , Jun. 1997, 36 -41.

[8] G.M. Megson, *An introduction to systolic algorithm design,* Oxford Science Publications, 1992.

[9] K.H.Hwang, Z.Xu, *Scalable parallel computing,* McGraw Hill, 1998.

[10] B. Faires, *Numerical Analysis, PWS*Kent Publishing Co., 1993.

[11] A. El-Amawy, A systolic architecture for fast dense matrix inversion, IEEE Trans. on Computer, 38(4), March 1989, 449-455.

[12] V. Zebre, H. Keller, G. Schorcht, Parallel Matrix Computation and their Applications for Biomagnetic Fields, IEEE Advanced in Parallel and Distributed Computing, APDC97, 1997, 139.